МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФГБОУ ВО «ИГУ»
Кафедра физико-химической биологии, биоинженерии и биоинформатики

УТВЕРЖДАЮ
Декан биолого-почвенного факультета
А.Н. Матвеев
«21» марта 2025 г.

ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

для проведения текущего контроля и промежуточной аттестации по дисциплине:

Б1.О.43 **«АЛГОРИТМЫ БИОИНФОРМАТИКИ»**

Специальность: 06.05.01 «Биоинженерия и биоинформатика»
Направленность (профиль): «Биоинженерия и биоинформатика»

Квалификация выпускника: биоинженер и биоинформатик

Форма обучения: очная с элементами электронного обучения и дистанционных образовательных технологий

Согласовано с УМК биолого-почвенного факультета
Протокол № 5 от 21 марта 2025 г.
Председатель _____ А.Н. Матвеев

Рекомендовано кафедрой физико-химической биологии, биоинженерии и биоинформатики
Протокол № 12 от 19 марта 2025 г.
Зав. кафедрой _____ В.П. Саловарова

Иркутск 2025 г.

ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

Разработан для учебной дисциплины Б1.О.43 «АЛГОРИТМЫ БИОИНФОРМАТИКИ» специальности 06.05.01 «Биоинженерия и биоинформатика», специализация «Биоинженерия и биоинформатика». Фонд оценочных материалов (ФОМ) включает оценочные материалы для проведения текущего контроля, промежуточной аттестации в форме зачета.

Оценочные материалы соотнесены с требуемыми результатами освоения образовательной программы 06.05.01 «Биоинженерия и биоинформатика», в соответствии с содержанием рабочей программы учебной дисциплины Б1.О.43 «Алгоритмы биоинформатики» с учетом ОПОП.

Нормативные документы, регламентирующие разработку ФОМ:

- статья 2, часть 9 Федерального закона «Об образовании в Российской Федерации», ФЗ-273, от 29.12.2012 г.;
- ФГОС ВО по специальности 06.05.01 «Биоинженерия и биоинформатика», утвержденный приказом Министерства науки и высшего образования Российской Федерации 12 августа 2020 г. № 973.

1. Компетенции, формируемые в процессе изучения дисциплины (4 курс, 7 семестр)

ОПК-6: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.
ОПК-7: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Компетенция	**Индикаторы компетенций**	**Результаты обучения**	**Формы и методы контроля и оценки**
ПК-1 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	*ИДК ОПК-6.1* Знает принципы создания компьютерных программ, используемых в биоинформатике и биоинженерии	Знать: литературу по теме, владеть навивками, анализа информации сети «интернет» для поиска и освоения новых методов анализа данных, информационных технологий и алгоритмов. Уметь: выбирать оптимальные методы, алгоритмы и программы для решения задач в области анализа биологической информации в геномики, эволюционной биологии и экологии. Владеть: методами построение сложных алгоритмов, с применением бутсреп метода, алгоритмов на основе показателей правдоподобия и машинного обучения.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: зачет
	ИДК ОПК-6.2 Использует современные IT-технологии при сборе, анализе, обработке и представлении информации	Знать: классификацию алгоритмов, основные типы алгоритмов, синтаксис в области бутсрепа, алгоритмов на основе показателей правдоподобия и машинного обучения. Уметь: анализировать входные и выходные данные разрабатываемого алгоритма,	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная**

		производить отладку и тестирование разработанных алгоритмов для анализа данных в эволюционной биологии, геномики и экологи. Владеть: навыками анализа сложных данных в различных отраслях биологии и биоинформатики.	**аттестация**: зачет
	ИДК ОПК-6.3 Использует навыки создания компьютерных программ, баз данных и иные программных продуктов, используемых в биоинженерии и биоинформатике	Знать: классификацию основных типов алгоритмов анализа сложных данных и применять изученные алгоритмы для создания сложных конвейеров анализа данных. Уметь: осуществлять интерпретацию результатов математических расчетов с применением всех изученных типов алгоритмов. Владеть: методами анализа комплексных биологических данных в эволюционной биологии, геномики и экологии	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: зачет
ОПК-7 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	*ИДК ОПК-7.1* Демонстрирует теоретические и практические навыки использования современных информационных технологий в области профессианальной деятельности	Знать: основные математические понятия и методы, применимые для анализа биологички систем и биологических данных с применением алгоритмов бустреп анализа алгоритмов на основе показателей правдоподобия и машинного обучения. Уметь: адекватно выбрать математический метод и алгоритмы для описания биологической системы и биологического процесса в эволюционной биологии геномики, биоинформатики и экологии. Владеть: основными принципами формализации сложных конвейеров анализа данных с применением всех рассмотренных алгоритмов.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: зачет
	ИДК ОПК-7.2 Использует современные информационные технологии в рамках освоения материала и реализации задач в области профессианальной деятельности	Знать: цель, основные задачи и области применения алгоритмов биоинформатики в рамках направления подготовки. Уметь: формализовать процесс обработки данных в геномики, эволюционной биологии, экологии и других биологических дисциплинах в виде конвейеров различных вычислительных алгоритмов. Владеть: методами применения разработанных алгоритмов и конвейеров анализа данных при исследовании биологических процессов и биосистем.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: зачет

2. Оценочные материалы текущего контроля

В рамках дисциплины «Алгоритмы биоинформатики» используются следующие формы текущего контроля - письменная работа по решению самостоятельных заданий (все формулировки заданий для самостоятельного решения с необходимыми сопроводительными материалами выложены на образовательном портале ИГУ в темах курса «Алгоритмы биоинформатики»);

Перечень письменных работ для самостоятельного выполнения по разделам – темам дисциплины.

Задание по теме 1:
Цель: Реализовать бутстреп-метод на языке R для оценки доверительного интервала среднего значения.
Описание:
Вам предоставляется выборка данных. Ваша задача - используя бутстреп, построить доверительный интервал для среднего значения этой выборки. Необходимо реализовать функцию, которая принимает выборку, количество бутстреп-выборок и уровень доверия, а возвращает нижнюю и верхнюю границы доверительного интервала.
Входные данные:
- `data`: Вектор числовых данных (выборка).
- `n_boot`: Количество бутстреп-выборок для ресэмплинга.
- `conf_level`: Уровень доверия (например, 0.95 для 95% доверительного интервала).

Выходные данные:
- Вектор, содержащий нижнюю и верхнюю границы доверительного интервала.

Рекомендации:
1. Создайте функцию `bootstrap_ci`, которая принимает входные данные, указанные выше.
2. Внутри функции генерируйте `n_boot` бутстреп-выборок путем случайного выбора с возвращением из исходной выборки `data`.
3. Для каждой бутстреп-выборки рассчитайте среднее значение.
4. Рассчитайте `alpha = 1 - conf_level`, и найдите `alpha/2` и `1 - alpha/2` квантили распределения бутстреп-средних. Эти квантили будут являться нижней и верхней границами доверительного интервала соответственно.
5. Верните результирующий вектор, содержащий нижнюю и верхнюю границы доверительного интервала.

Ответ:

```
bootstrap_ci <- function(data, n_boot, conf_level) {
  # 1. Инициализация
  n <- length(data)
  boot_means <- numeric(n_boot) # Вектор для хранения средних бутстреп-выборок

  # 2. Генерация бутстреп-выборок и расчет средних
  for (i in 1:n_boot) {
    boot_sample <- sample(data, size = n, replace = TRUE) # выборка с возвращением
    boot_means[i] <- mean(boot_sample) # расчет среднего
  }

  # 3. Расчет квантилей для доверительного интервала
  alpha <- 1 - conf_level
  lower_quantile <- alpha / 2
  upper_quantile <- 1 - alpha / 2

  lower_bound <- quantile(boot_means, lower_quantile)
```

```
  upper_bound <- quantile(boot_means, upper_quantile)

  # 4. Возврат результата
  return(c(lower_bound, upper_bound))
}

# Пример использования (как в задании)
data <- rnorm(100, mean = 5, sd = 2) # Пример выборки из нормального распределения
n_boot <- 1000
conf_level <- 0.95

ci <- bootstrap_ci(data, n_boot, conf_level)
print(ci)
```

Задание по теме 2:

Цель: Реализовать бутстреп-метод на языке R для проверки гипотезы о наличии значимого различия между средними значениями двух независимых выборок.

Описание:

Вам даны две независимые выборки. Ваша задача - используя бутстреп в рамках подхода перестановочного теста (permutation test), протестировать нулевую гипотезу о том, что средние значения двух популяций, из которых взяты выборки, равны (H0: μ1 = μ2). Альтернативная гипотеза - средние значения различны (H1: μ1 ≠ μ2).

Реализуйте функцию, которая принимает две выборки, определяет тестовую статистику (разность средних), выполняет бутстреп-перестановки, вычисляет p-value и возвращает результат теста (отвергаем или не отвергаем H0).

Входные данные:

- `sample1`: Вектор числовых данных (первая выборка).
- `sample2`: Вектор числовых данных (вторая выборка).
- `n_permutations`: Количество бутстреп-перестановок.

Выходные данные:

- Строка, сообщающая результат теста: "Отвергаем H0" или "Не отвергаем H0".
- Значение p-value.

Рекомендации:

1. Создайте функцию `bootstrap_hypothesis_test`, которая принимает входные данные, указанные выше.
2. Вычислите наблюдаемую разность средних – тестовую статистику – между двумя выборками: `observed_diff <- mean(sample1) - mean(sample2)`.
3. Объедините две выборки в одну общую выборку: `combined_sample <- c(sample1, sample2)`.
4. Внутри функции генерируйте `n_permutations` бутстреп-перестановок данных. Для каждой перестановки:
   - Случайным образом перетасуйте объединенную выборку.
   - Разделите перетасованную выборку на две новые выборки так же, как и исходные (первые `length(sample1)` элементов – первая перетасованная выборка, остальные – вторая).
   - Вычислите разность средних для полученных перетасованных выборок.
5. Вычислите p-value. P-value – это пропорция перестановок, при которых абсолютное значение разности средних в перетасованных выборках больше или равно абсолютному значению наблюдаемой разности средних. Используйте: `p_value <- mean(abs(boot_diffs) >= abs(observed_diff))`.
6. Задайте уровень значимости `alpha` (например, 0.05).
7. Примите решение об отвержении или неотвержении нулевой гипотезы: если `p_value <= alpha`, отвергаем H0; иначе не отвергаем H0.
8. Верните строку с результатом теста и значение p-value.

Ответ:

```
bootstrap_hypothesis_test <- function(sample1, sample2, n_permutations) {
```

```
  # 1. Вычисление наблюдаемой разности средних
  observed_diff <- mean(sample1) - mean(sample2)

  # 2. Объединение выборок
  combined_sample <- c(sample1, sample2)

  # 3. Инициализация вектора для хранения разностей средних в перестановках
  boot_diffs <- numeric(n_permutations)

  # 4. Бутстреп-перестановки
  for (i in 1:n_permutations) {
    # Перемешивание объединенной выборки
    shuffled_sample <- sample(combined_sample, size = length(combined_sample), replace = FALSE)

    # Разделение на псевдо-выборки
    boot_sample1 <- shuffled_sample[1:length(sample1)]
    boot_sample2 <- shuffled_sample[(length(sample1) + 1):length(combined_sample)]

    # Вычисление разности средних для перестановки
    boot_diffs[i] <- mean(boot_sample1) - mean(boot_sample2)
  }

  # 5. Вычисление p-value
  p_value <- mean(abs(boot_diffs) >= abs(observed_diff))

  # 6. Принятие решения
  alpha <- 0.05  # Уровень значимости
  if (p_value <= alpha) {
    decision <- "Отвергаем H0"
  } else {
    decision <- "Не отвергаем H0"
  }

  # 7. Возврат результата
  return(list(decision = decision, p_value = p_value))
}

# Пример использования (как в задании)
sample1 <- rnorm(30, mean = 10, sd = 3)
sample2 <- rnorm(30, mean = 8, sd = 3)
n_permutations <- 10000

result <- bootstrap_hypothesis_test(sample1, sample2, n_permutations)
print(result)
```

Задание по теме 3:

Цель: Реализовать бутстреп-метод в R для оценки доверительного интервала коэффициента корреляции Пирсона.

Описание:

Вам даны два вектора данных, представляющих собой парные наблюдения двух переменных. Ваша задача – использовать бутстреп для построения доверительного интервала для

коэффициента корреляции Пирсона между этими переменными. Реализуйте функцию, которая принимает два вектора данных, количество бутстреп-выборок и требуемый уровень доверия, вычисляет коэффициент корреляции Пирсона для каждой бутстрепированной выборки и возвращает нижнюю и верхнюю границы доверительного интервала.

Входные данные:

- `x`: Вектор числовых данных (первая переменная).
- `y`: Вектор числовых данных (вторая переменная). Убедитесь, что `length(x) == length(y)`.
- `n_boot`: Количество бутстреп-выборок.
- `conf_level`: Уровень доверия (например, 0.95 для 95% доверительного интервала).

Выходные данные:

- Вектор, содержащий нижнюю и верхнюю границы доверительного интервала для коэффициента корреляции Пирсона.

Рекомендации:

1. Создайте функцию `bootstrap_correlation_ci`, которая принимает входные данные, указанные выше.
2. Внутри функции сделайте проверку: если длина векторов x и y не совпадает, функция должна возвращать ошибку или NULL с сообщением об ошибке.
3. Сформируйте "пары" из x и y для упрощения ресэмплинга.
4. Для каждой бутстреп-выборки (от 1 до `n_boot`):
   - Выполните случайный выбор с возвращением пар (индексов) из исходных данных.
   - Используйте выбранные индексы для создания новых векторов `x` и `y` (бутстреп-выборки).
   - Вычислите коэффициент корреляции Пирсона между новыми `x` и `y` и сохраните его.
5. Рассчитайте квантили распределения бутстреп-коэффициентов корреляции, соответствующие `alpha/2` и `1 - alpha/2`, где `alpha = 1 - conf_level`. Эти квантили будут являться нижней и верхней границами доверительного интервала.
6. Верните результирующий вектор, содержащий нижнюю и верхнюю границы доверительного интервала

Ответ:

```
bootstrap_correlation_ci <- function(x, y, n_boot, conf_level) {
 # 1. Проверка входных данных
 if (length(x) != length(y)) {
  stop("Длины векторов x и y должны совпадать.")  # Обработка ошибки
 }

 n <- length(x) # размер выборки
 boot_correlations <- numeric(n_boot) # Вектор для хранения бутстреп-коэффициентов корреляции

 # 2. Бутстреп-цикл
 for (i in 1:n_boot) {
  # Случайный выбор индексов с возвращением
  indices <- sample(1:n, size = n, replace = TRUE)

  # Создание бутстреп-выборок
  boot_x <- x[indices]
  boot_y <- y[indices]

  # Вычисление коэф. корреляции Пирсона и сохранение
  boot_correlations[i] <- cor(boot_x, boot_y)
 }

 # 3. Расчет квантилей для доверительного интервала
```

```
  alpha <- 1 - conf_level
  lower_quantile <- alpha / 2
  upper_quantile <- 1 - alpha / 2

  lower_bound <- quantile(boot_correlations, lower_quantile)
  upper_bound <- quantile(boot_correlations, upper_quantile)

  # 4. Возврат результата
  return(c(lower_bound, upper_bound))
}

# Пример использования (как в задании)
x <- rnorm(100)
y <- x + rnorm(100, sd = 0.5) # Зависимые переменные
n_boot <- 1000
conf_level <- 0.95

ci <- bootstrap_correlation_ci(x, y, n_boot, conf_level)
print(ci)
```

Задание по теме 4:

Цель: Реализовать перестановочный тест в R для однофакторного дисперсионного анализа (ANOVA).

Описание:

Вам даны данные, представляющие собой измерения некоторой переменной в нескольких группах (факторах). Ваша задача — использовать перестановочный тест, чтобы проверить нулевую гипотезу о том, что средние значения переменной одинаковы во всех группах (H0: μ1 = μ2 = … = μk). Альтернативная гипотеза — по крайней мере одно среднее значение отличается (H1: хотя бы одно μi ≠ μj).

Реализуйте функцию, которая принимает данные, формирует тестовую статистику (F-статистику), выполняет перестановки, вычисляет p-value и возвращает результат теста.

Входные данные:

- `data`: Список (list) или таблица данных (data.frame). Если `data` - это list, то каждый элемент списка является вектором числовых данных, представляющим измерения для одной группы. Если `data` - data.frame, то она должна содержать два столбца: один с измерениями, другой с идентификатором группы (фактором).
- `n_permutations`: Количество перестановок.

Выходные данные:

- Строка, сообщающая результат теста: "Отвергаем H0" или "Не отвергаем H0".
- Значение p-value.
- Значение наблюдаемой F-статистики.

Рекомендации:

1. Создайте функцию `permutation_anova`, которая принимает входные данные, указанные выше.
2. Реализуйте функцию для вычисления F-статистики. F-статистика в ANOVA измеряет вариацию между группами относительно вариации внутри групп. (Сумма квадратов между группами/степени свободы между группами) / (Сумма квадратов внутри групп/степени свободы внутри групп). Вам понадобится функция для подсчета суммы квадратов (sum of squares).
3. Рассчитайте наблюдаемую F-статистику для исходных данных.
4. Объедините все данные из разных групп в один вектор.
5. Внутри функции генерируйте `n_permutations` перестановок данных. Для каждой перестановки:
   - Случайным образом перетасуйте объединенные данные.
   - Разделите перетасованные данные обратно на группы, сохраняя исходные размеры групп.

- Вычислите F-статистику для полученных перетасованных групп.

6. Вычислите p-value. P-value – это доля перестановок, при которых F-статистика в перетасованных данных больше или равна наблюдаемой F-статистике. Используйте: `p_value <- mean(boot_F >= observed_F)`.
7. Задайте уровень значимости `alpha` (например, 0.05).
8. Примите решение об отвержении или неотвержении нулевой гипотезы: если `p_value <= alpha`, отвергаем H0; иначе не отвергаем H0.
9. Верните строку с результатом теста, значение p-value и наблюдаемую F-статистику

Ответ:

```
# Функция для расчета F-статистики
calculate_f_statistic <- function(data, group) {
  # Общее среднее
  grand_mean <- mean(data)

  # Количество групп
  k <- length(unique(group))

  # Сумма квадратов между группами (SSB)
  ssb <- sum(tapply(data, group, function(x) {
    (mean(x) - grand_mean)^2 * length(x)
  }))

  # Сумма квадратов внутри групп (SSW)
  ssw <- sum(tapply(data, group, function(x) {
    sum((x - mean(x))^2)
  }))

  # Степени свободы
  dfb <- k - 1
  dfw <- length(data) - k

  # Средний квадрат между группами (MSB)
  msb <- ssb / dfb

  # Средний квадрат внутри групп (MSW)
  msw <- ssw / dfw

  # F-статистика
  f_statistic <- msb / msw

  return(f_statistic)
}

permutation_anova <- function(data, n_permutations) {

  # Обработка входных данных (поддержка list и data.frame)
  if (is.list(data)) {
    # list формат
    values <- unlist(data) # объединение списков в один большой вектор.  Потеря групп!
    groups <- factor(rep(names(data), sapply(data, length))) # создаем фактор с идентификаторами групп
  } else if (is.data.frame(data)) {
```

```
  # data.frame формат.  Предполагаем колонки "measurement" и "group"
  if (!all(c("measurement", "group") %in% colnames(data))) {
    stop("Data frame должен содержать колонки 'measurement' и 'group'.")
  }
  values <- data$measurement
  groups <- data$group
 } else {
  stop("Неподдерживаемый формат входных данных.  Используйте list или data.frame.")
 }


 # 1. Расчет наблюдаемой F-статистики
 observed_F <- calculate_f_statistic(values, groups)

 # 2. Инициализация вектора для хранения F-статистик для перестановок
 boot_F <- numeric(n_permutations)

 # 3. Цикл перестановок
 for (i in 1:n_permutations) {
  # Перемешивание данных
  shuffled_groups <- sample(groups)

  # Вычисление F-статистики для перестановки
  boot_F[i] <- calculate_f_statistic(values, shuffled_groups)
 }

 # 4. Вычисление p-value
 p_value <- mean(boot_F >= observed_F)

 # 5. Принятие решения
 alpha <- 0.05  # Уровень значимости
 if (p_value <= alpha) {
  decision <- "Отвергаем H0"
 } else {
  decision <- "Не отвергаем H0"
 }

 # 6. Возврат результата
 return(list(decision = decision, p_value = p_value, observed_F = observed_F))
}


# Пример использования (list)
data <- list(group1 = rnorm(20, mean = 10, sd = 2),
             group2 = rnorm(20, mean = 12, sd = 2),
             group3 = rnorm(20, mean = 11, sd = 2))
n_permutations <- 10000

result <- permutation_anova(data, n_permutations)
print(result)

# Пример использования (data.frame)
```

```
measurement <- c(rnorm(20, mean = 10, sd = 2), rnorm(20, mean = 12, sd = 2), rnorm(20, mean = 11, sd = 2))
group <- factor(rep(c("A", "B", "C"), each = 20))
data <- data.frame(measurement = measurement, group = group)

n_permutations <- 10000
result <- permutation_anova(data, n_permutations)
print(result)
```

Задание по теме 5:

Цель: Реализовать оценку параметров нормального распределения (математического ожидания μ и стандартного отклонения σ) методом максимального правдоподобия (MLE) в R.

Описание:

Вам дана выборочная совокупность, предположительно полученная из нормального распределения. Ваша задача — реализовать функцию, вычисляющую оценки для параметров μ и σ с использованием метода максимального правдоподобия.

Входные данные:

- `data`: Вектор, содержащий выборочные данные (числовые значения).

Выходные данные:

- Список, содержащий два элемента:
  - `mu`: Оценка математического ожидания (μ).
  - `sigma`: Оценка стандартного отклонения (σ).

Подсказки:

- Функция правдоподобия для нормального распределения:
  $L(\mu, \sigma \mid x) = \prod (1 / (\sigma * \sqrt{2\pi})) * \exp(-(x_i - \mu)^2 / (2\sigma^2))$
  где:
  - x - вектор данных.
  - μ - математическое ожидание.
  - σ - стандартное отклонение.
  - ∏ означает произведение всех элементов.
- Логарифмическая функция правдоподобия: Более удобно максимизировать логарифм функции правдоподобия, так как это упрощает вычисления (произведение превращается в сумму) и не меняет положение максимума.
  $\log L(\mu, \sigma \mid x) = \Sigma [-\log(\sigma) - \log(\sqrt{2\pi}) - (x_i - \mu)^2 / (2\sigma^2)]$
- Чтобы найти оценки MLE для μ и σ, нужно найти значения, которые максимизируют логарифмическую функцию правдоподобия. Для простоты, можно использовать функцию `optim()` в R для численной оптимизации. Альтернативно, можно аналитически вывести формулы для оценок MLE (что является более сложным, но поучительным подходом). В данном задании ожидается использование `optim()`.
- Формулы для оценок MLE (аналитическое решение - предоставляется для справки, но не обязательно для реализации, так как используется `optim()`):
  - $\hat{\mu} = (1/n) * \Sigma x_i$ (выборочное среднее)
  - $\hat{\sigma}^2 = (1/n) * \Sigma (x_i - \hat{\mu})^2$ (выборочная дисперсия со знаменателем *n*, то есть смещенная оценка)
  - $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$

Рекомендации:

1. Создайте функцию `mle_normal`, которая принимает вектор данных `data` в качестве аргумента.
2. Определите логарифмическую функцию правдоподобия (`log_likelihood`). Эта функция должна принимать параметры `mu` и `sigma`, а также данные `data`, и возвращать значение логарифмической функции правдоподобия (отрицательное значение, так как `optim` по умолчанию *минимизирует*). Убедитесь, что стандартное отклонение (`sigma`) всегда положительное (например, можно использовать `exp(log_sigma)` внутри функции). В R `dnorm(x, mean = mu, sd = sigma, log = TRUE)` возвращает логарифм плотности нормального распределения для x, что упрощает написание логарифмической функции правдоподобия.

3. Используйте функцию `optim()` для максимизации логарифмической функции правдоподобия. Ей нужно передать начальные оценки параметров μ (например, выборочное среднее) и σ (например, выборочное стандартное отклонение), логарифмическую функцию правдоподобия, данные. Важно обратить внимание на то, что `optim()` *минимизирует* функцию, поэтому используйте *отрицательную* логарифмическую функцию правдоподобия.
4. Извлеките оценки для μ и σ из результата работы `optim()`.
5. Верните список, содержащий оценки для `mu` и `sigma`.

Ответ:

```
mle_normal <- function(data) {

 # 1.  Логарифмическая функция правдоподобия (используем dnorm() для простоты)
 log_likelihood <- function(params, data) {
  mu <- params[1]
  log_sigma <- params[2] # Использование log_sigma чтобы обеспечить sigma > 0
  sigma <- exp(log_sigma)

  # Сумма логарифмов плотностей нормального распределения
  log_likelihood_values <- dnorm(data, mean = mu, sd = sigma, log = TRUE)
  sum_log_likelihood <- sum(log_likelihood_values)

  return(-sum_log_likelihood) # Возвращаем отрицательное значение, так как optim минимизирует
 }

 # 2. Начальные оценки для параметров
 mu_initial <- mean(data)
 sigma_initial <- sd(data)
 log_sigma_initial <- log(sigma_initial) # Начальная оценка для log_sigma

 # 3. Оптимизация с помощью optim()
 optimization_result <- optim(par = c(mu_initial, log_sigma_initial),  # Начальные значения
                              fn = log_likelihood, # Функция для минимизации (отрицательный логарифм правдоподобия)
                              data = data,        # Данные
                              hessian = FALSE)     # Вычисление матрицы Гессе не требуется

 # 4. Извлечение оценок параметров
 mu_mle <- optimization_result$par[1]
 sigma_mle <- exp(optimization_result$par[2]) # Получаем sigma из log_sigma

 # 5. Возврат результата
 return(list(mu = mu_mle, sigma = sigma_mle))
}

# Пример использования:
data <- rnorm(100, mean = 5, sd = 2) # Генерируем случайные данные из нормального распределения
result <- mle_normal(data)
print(result) # Вывод: что-то близкое к  mu = 5 и sigma = 2
```

Задание по теме 6:

Цель: Реализовать алгоритм MCMC (Markov Chain Monte Carlo) для оценки параметров линейной регрессии в R.

Описание:

Вам дан набор данных для линейной регрессии. Ваша задача — реализовать MCMC алгоритм для оценки параметров линейной регрессии: коэффициентов регрессии (β) и стандартного отклонения ошибки (σ). Предположим, что модель имеет вид:

$y = X\beta + \varepsilon$, где $\varepsilon \sim N(0, \sigma^2)$

Предположим, что у нас есть только один предиктор (одна независимая переменная) для простоты. То есть, X - это матрица, содержащая столбец единиц (для свободного члена регрессии) и столбец значений предиктора. β состоит из двух параметров: β_0 (свободный член) и β_1.

В качестве априорных распределений используйте:

- β_0 и β_1: нормальное распределение с $\mu = 0$ и $\sigma = 10$.
- σ: полунормальное распределение (т.е. $\sigma > 0$) с параметром масштаба $\sigma = 5$.

Используйте алгоритм Metropolis-Hastings для генерации выборок из апостериорного распределения параметров.

Входные данные:

- `y`: Вектор зависимой переменной.
- `x`: Вектор независимой переменной (предиктора).
- `iterations`: Количество итераций MCMC.

Выходные данные:

- Матрица `samples`, где каждая строка представляет собой выборку из апостериорного распределения, а столбцы соответствуют:
  - `beta_0`: Оценка коэффициента β_0 (свободного члена).
  - `beta_1`: Оценка коэффициента β_1.
  - `sigma`: Оценка стандартного отклонения ошибки σ.
- `accept_rate`: Доля принятых значений.

Рекомендации:

1. Создайте функцию `mcmc_linear_regression`, которая принимает входные данные, указанные выше.
2. **Функция логарифма апостериорного распределения:** Определите функцию `log_posterior`, которая вычисляет логарифм апостериорного распределения параметров, заданных как `params`:
   - `params[1]` == `beta_0`
   - `params[2]` == `beta_1`
   - `params[3]` == `sigma` Функция берет на входе (y, X, params). Вам потребуется вычислить:
   - Логарифм правдоподобия (log-likelihood) данных, учитывая параметры и модель. Предполагается нормальное распределение ошибок.
   - Логарифм априорного распределения для β_0 и β_1 (нормальное распределение).
   - Логарифм априорного распределения для σ (полунормальное распределение).
   - Логарифм апостериорного распределения получается путем сложения всех этих логарифмов.
3. **Инициализация:** Инициализируйте цепь MCMC, выбрав начальные значения для параметров (например, $\beta_0 = 0$, $\beta_1 = 0$, $\sigma = 1$).
4. **Цикл MCMC:** Для каждой итерации:
   - **Предложение:** Сгенерируйте новые значения параметров из функции предложения (proposal distribution). Простая стратегия - добавить случайный шум из нормального распределения к текущим значениям параметров. Рекомендуется использовать `rnorm` для генерации случайного шума.
   - **Оценка:** Вычислите логарифм апостериорного распределения для текущих значений параметров и для предложенных значений.
   - **Принятие/отклонение:** Вычислите отношение правдоподобия (likelihood ratio) или отношение Metropolis-Hastings. Сгенерируйте случайное число из равномерного распределения между 0 и 1. Если это число меньше отношения правдоподобия, примите предложенные значения параметров. В противном случае, отклоните предложение и сохраните текущие значения параметров. Обратите внимание, что

нужно работать с логарифмами. Рассчитайте вероятность принятия (acceptance probability) как `min(1, exp(log_posterior_new - log_posterior_current))`.
   - **Сохранение:** Сохраните принятые значения параметров в матрицу `samples`.
5. **Функции предложения:** Используйте нормальное распределение для "предложения" новых значений параметров. Важно, чтобы стандартное отклонение функции предложения было подобрано правильно, чтобы обеспечить адекватную скорость принятия (acceptance rate).
6. **Адаптация (необязательно, но рекомендуется):** В процессе выполнения MCMC алгоритма адаптируйте стандартное отклонение функции предложения для каждого параметра, чтобы добиться оптимальной скорости принятия. Оптимальная скорость принятия обычно составляет около 20-40%.
7. **Возврат результата:** Верните матрицу `samples`, содержащую значения параметров, полученные в результате MCMC, и `accept_rate`.

Ответ:

```
mcmc_linear_regression <- function(y, x, iterations) {

 # 1. Подготовка данных (матрица X)
 X <- cbind(1, x)  # Добавляем столбец единиц для свободного члена
 n <- length(y)

 # 2. Функция логарифма апостериорного распределения
 log_posterior <- function(params, y, X) {
  beta_0 <- params[1]
  beta_1 <- params[2]
  sigma <- params[3]

  # Проверка на допустимые значения (sigma > 0)
  if (sigma <= 0) return(-Inf)

  beta <- c(beta_0, beta_1)

  # 2.1. Логарифм правдоподобия
  predictions <- X %*% beta
  log_likelihood <- sum(dnorm(y, mean = predictions, sd = sigma, log = TRUE))

  # 2.2. Логарифм априорного распределения для beta_0 и beta_1 (нормальное)
  log_prior_beta <- dnorm(beta_0, mean = 0, sd = 10, log = TRUE) +
                    dnorm(beta_1, mean = 0, sd = 10, log = TRUE)

  # 2.3. Логарифм априорного распределения для sigma (полунормальное)
  log_prior_sigma <- dnorm(sigma, mean = 0, sd = 5, log = TRUE) # Замечание: это "обрезанное" нормальное, де-факто

  # 2.4. Логарифм апостериорного распределения
  log_posterior <- log_likelihood + log_prior_beta + log_prior_sigma

  return(log_posterior)
 }

 # 3. Инициализация
 beta_0_start <- 0
 beta_1_start <- 0
 sigma_start <- 1
```

```
  params_current <- c(beta_0_start, beta_1_start, sigma_start)

  # 4. Настройка MCMC
  samples <- matrix(NA, nrow = iterations, ncol = 3)
  colnames(samples) <- c("beta_0", "beta_1", "sigma")
  samples[1, ] <- params_current

  # Параметры функции предложения (proposal distribution) - нужно настраивать!
  proposal_sd <- c(0.5, 0.5, 0.2) # Стандартные отклонения нормального распределения

  # Переменные для отслеживания скорости принятия
  accepted <- 0

  # 5. Цикл MCMC
  for (i in 2:iterations) {
    # 5.1. Предложение новых значений
    params_proposal <- rnorm(3, mean = params_current, sd = proposal_sd)

    # 5.2. Вычисление логарифма апостериорного распределения для текущих и предложенных значений
    log_posterior_current <- log_posterior(params_current, y, X)
    log_posterior_proposal <- log_posterior(params_proposal, y, X)

    # 5.3. Вычисление вероятности принятия
    acceptance_probability <- min(1, exp(log_posterior_proposal - log_posterior_current))

    # 5.4. Принятие или отклонение
    if (runif(1) < acceptance_probability) {
      params_current <- params_proposal
      accepted <- accepted + 1 # Увеличиваем счетчик принятых значений
    }

    # 5.5. Сохранение результатов
    samples[i, ] <- params_current
  }

  # 6. Вычисление скорости принятия
  accept_rate <- accepted / iterations

  # 7. Возврат результата
  return(list(samples = samples, accept_rate = accept_rate))
}


# Пример использования:
# Пример данных
set.seed(123)
x <- rnorm(100)
y <- 2 + 3*x + rnorm(100, 0, 2) # y = 2 + 3x + error

# Запуск MCMC
iterations <- 10000
```

```
result <- mcmc_linear_regression(y, x, iterations)

# Анализ результатов
samples <- result$samples
accept_rate <- result$accept_rate

print(paste("Acceptance Rate:", accept_rate))

# Вывод средних значений параметров
print(colMeans(samples))

# Визуализация распределений параметров
par(mfrow=c(1,3))
hist(samples[, "beta_0"], main="Beta 0", xlab="")
hist(samples[, "beta_1"], main="Beta 1", xlab="")
hist(samples[, "sigma"], main="Sigma", xlab="")
```

Задание по теме 7:

Цель: Разработать функцию в R, которая генерирует случайные числа из распределения Пуассона, где параметр λ (среднее значение) сам случайно берется из нормального распределения.

Описание:

Обычное распределение Пуассона характеризуется одним параметром λ (лямбда), который определяет среднее значение и дисперсию распределения. В этом задании мы усложняем задачу, предполагая, что параметр λ не является фиксированным, а сам распределен по нормальному закону. Таким образом, мы получим составное распределение, где:

1. **$\lambda \sim N(\mu, \sigma^2)$**: Параметр λ распределен нормально со средним μ и дисперсией σ^2. Важно отметить, что λ должно быть положительным. Мы можем получить положительные значения, используя абсолютную величину нормального распределения (то есть, $|N(\mu, \sigma^2)|$). Внимание: это делает математические свойства распределения более сложными (это уже не стандартное распределение, а его модификация).
2. **$X \sim Poisson(\lambda)$**: Для каждого сгенерированного значения λ генерируется случайное число X из распределения Пуассона с параметром λ.

Задание:

1. Разработайте функцию на языке R `rpoisnorm(n, mu, sigma)`, которая генерирует `n` случайных чисел из указанного выше составного распределения Пуассона-Нормального. Функция должна принимать следующие аргументы:
   - `n`: Количество случайных чисел для генерации.
   - `mu`: Среднее значение нормального распределения для параметра λ.
   - `sigma`: Стандартное отклонение нормального распределения для параметра λ.
2. Внутри функции `rpoisnorm`:
   - Сгенерируйте `n` случайных значений λ из нормального распределения с параметрами `mu` и `sigma`. Используйте функцию `rnorm()`.
   - Обеспечьте, чтобы все значения λ были положительными, взяв абсолютное значение полученных случайных чисел из нормального распределения.
   - Для каждого сгенерированного значения λ сгенерируйте случайное число из распределения Пуассона с параметром λ. Используйте функцию `rpois()`.
   - Верните вектор из `n` случайных чисел, сгенерированных из распределения Пуассона.
3. Проверьте разработанную функцию:
   - Сгенерируйте 1000 случайных чисел с `mu = 5` и `sigma = 2`.
   - Постройте гистограмму сгенерированных чисел.
   - Вычислите среднее значение и дисперсию сгенерированных чисел и сравните их с ожидаемыми значениями (теоритечески, среднее значение должно быть около `mu`, а дисперсию надо считать аналитически или оценивать моделированием).

Рекомендации:

- Убедитесь, что функция обрабатывает корректно случай, когда `sigma` близко к нулю (в этом случае, все значения λ будут близки к `mu`).
- Подумайте о том, как можно оптимизировать функцию (например, векторизация операций).

Оценка:

- Корректная реализация функции `rpoisnorm`.
- Правильная генерация случайных чисел из нормального и Пуассоновского распределений.
- Реализация требования положительности параметра λ.
- Проверка и валидация сгенерированных данных.
- Чистый и хорошо задокументированный код.

Ответ:

```
rpoisnorm <- function(n, mu, sigma) {
  # Генерируем n случайных значений lambda из нормального распределения
  lambda <- rnorm(n, mean = mu, sd = sigma)

  # Берем абсолютное значение, чтобы lambda было положительным
  lambda <- abs(lambda)

  # Генерируем n случайных чисел из распределения Пуассона с параметрами lambda
  x <- rpois(n, lambda = lambda)

  # Возвращаем вектор случайных чисел
  return(x)
}

# Проверка функции
# Задаем параметры
n <- 1000
mu <- 5
sigma <- 2

# Генерируем случайные числа
random_numbers <- rpoisnorm(n, mu, sigma)

# Построение гистограммы
hist(random_numbers, main = "Гистограмма случайных чисел из составного распределения
Пуассона-Нормального",
     xlab = "Значение", ylab = "Частота", col = "skyblue", border = "white")

# Вычисление среднего значения и дисперсии
mean_value <- mean(random_numbers)
variance_value <- var(random_numbers)

# Вывод результатов
cat("Среднее значение:", mean_value, "\n")
cat("Дисперсия:", variance_value, "\n")

# Теоретические значения (приблизительно)
cat("Приблизительное теоретическое среднее:", mu, "\n")

# Чтобы посчитать тероритическую дисперсию нужно больше математики,
# так как используется абсолютное значение нормального распределения.
```

```
# В данном случае, дисперсия должна быть, приблизительно, больше чем mu
# из-за того что lambda распределена нормально и имеет дисперсию sigma^2.
```

Задание по теме 8:

Цель: Разработать эффективный алгоритм для поиска заданного нуклеотидного мотива в длинной нуклеотидной последовательности с использованием языка программирования R.

Описание:

В биоинформатике часто требуется находить определенные нуклеотидные мотивы (короткие последовательности ДНК) в больших геномных или транскриптомных последовательностях. В этом задании вам предлагается разработать функцию на R, которая находит все позиции заданного 12-буквенного мотива в нуклеотидной последовательности длиной 6,000,000 пар нуклеотидов.

Задание:

1. **Разработка функции `find_motifs(sequence, motif)`:**
   Разработайте функцию на языке R под названием `find_motifs(sequence, motif)`, которая принимает два аргумента:
   - `sequence`: Нуклеотидная последовательность (строка), в которой нужно искать мотив.
   - `motif`: Нуклеотидный мотив (строка), который нужно найти. Имеет длину 12 символов.

   Функция должна возвращать вектор, содержащий все начальные позиции, где мотив встречается в последовательности (индексация начинается с 1). Если мотив не найден, функция должна вернуть пустой вектор.
2. **Требования к алгоритму:**
   - Алгоритм должен быть эффективным и способным обрабатывать последовательность длиной 6,000,000 нуклеотидов за разумное время (не более нескольких минут).
   - Должны поддерживаться только стандартные нуклеотидные символы: A, T, G, и C. Функция должна возвращать ошибку, если последовательность или мотив содержат какие-либо другие символы.
   - Алгоритм должен возвращать ***все*** перекрывающиеся экземпляры мотива.
3. **Тестирование функции:**
   - Создайте случайную нуклеотидную последовательность длиной 6,000,000 нуклеотидов.
   - Выберите 12-буквенный мотив.
   - Используйте функцию `find_motifs` для поиска мотива в последовательности.
   - Убедитесь, что функция правильно находит все экземпляры мотива, включая перекрывающиеся.
   - Проверьте скорость работы функции.
4. **Оптимизация (бонус):**
   - Попробуйте оптимизировать функцию, чтобы достичь максимальной скорости. Можно рассмотреть варианты использования векторизации, функций из пакета `stringr`, или других методов.
   - Сравните время выполнения различных версий алгоритма

Ответ:

```
find_motifs <- function(sequence, motif) {
 # Проверка на допустимые символы
 valid_chars <- c("A", "T", "G", "C")
 seq_chars <- unique(unlist(strsplit(sequence, "")))
 motif_chars <- unique(unlist(strsplit(motif, "")))

 if (!all(seq_chars %in% valid_chars) || !all(motif_chars %in% valid_chars)) {
  stop("Последовательность и мотив должны содержать только символы A, T, G и C.")
 }

 # Проверка длины мотива
 if (nchar(motif) != 12) {
  stop("Длина мотива должна быть 12.")
 }
```

```
  # Длина последовательности и мотива
  seq_len <- nchar(sequence)
  motif_len <- nchar(motif)

  # Вектор для хранения позиций
  positions <- integer()

  # Поиск мотива в последовательности
  for (i in 1:(seq_len - motif_len + 1)) {
    if (substr(sequence, i, i + motif_len - 1) == motif) {
      positions <- c(positions, i)
    }
  }

  return(positions)
}

# --- Пример использования и тестирование ---

# Создание случайной нуклеотидной последовательности
set.seed(123) # Для воспроизводимости

generate_random_sequence <- function(length) {
  nucleotides <- c("A", "T", "G", "C")
  paste(sample(nucleotides, length, replace = TRUE), collapse = "")
}

sequence_length <- 6000000
random_sequence <- generate_random_sequence(sequence_length)

# Выбор мотива
motif <- "GCTAGCTAGCTA"  # Изменил на 12 символов.

# Замер времени выполнения
start_time <- Sys.time()
positions <- find_motifs(random_sequence, motif)
end_time <- Sys.time()

# Вывод результатов
cat("Мотив найден в позициях:\n")
print(positions)
cat("Количество найденных мотивов:", length(positions), "\n")

time_taken <- end_time - start_time
cat("Время выполнения:", time_taken, " секунд\n")


# --- Пример с короткими строками ---
sequence <- "ATTAGCTAGCTAGCTAGCT"
motif <- "GCTAGCTAGC"
positions <- find_motifs(sequence, motif)
```

```
print(positions)
# Test:  Expecte output positions = [6 11]

sequence <- "ATTAGCTAGCTAGC"
motif <- "GCTAGCTAGC"
positions <- find_motifs(sequence, motif)
print(positions)
# Test 2: Expecte output positions = [6]


# --- Примеры обработки ошибок ---
# Недопустимые символы
# find_motifs("ATGCX", "ATGC") # Error: Последовательность и мотив должны содержать только символы A, T, G и C.

# Некорректная длина мотива
# find_motifs("ATGC", "ATG")  # Error: Длина мотива должна быть 12.
# find_motifs("ATGC", "ATGCATGCAAGTTT") # Error: Длина мотива должна быть 12.
```

Задание по теме 9:

Цель: Научиться проводить функциональную аннотацию аминокислотных последовательностей, используя KofamKOALA (KEGG Orthology And Links Annotation) через командную строку Linux.

Описание:

Функциональная аннотация позволяет определить возможные функции, выполняемые белками, на основе их аминокислотных последовательностей. KofamKOALA - это инструмент, предназначенный для быстрой и точной аннотации геномов на основе базы данных KEGG Orthology (KO). В этом задании вы научитесь использовать KofamKOALA через командную строку Linux для аннотации набора аминокислотных последовательностей в формате FASTA.

Предварительные требования:

- Установленная система Linux (виртуальная машина подойдет).
- Установленный KofamKOALA. Инструкции по установке доступны на : https://www.genome.jp/kofamkoala/ и включают установку пакетов HMMER и KOALA. Важно установить необходимые профили KEGG/KO.
- Базовые знания командной строки Linux.
- Умение работать с форматом FASTA.

Задание:

1. **Подготовка FASTA файла:**
   - Создайте файл с аминокислотными последовательностями в формате FASTA. Файл должен содержать, по крайней мере, 5-10 последовательностей. Вы можете использовать примеры последовательностей из открытых баз данных (например, UniProt). Сохраните файл под именем `proteins.fasta`.
2. **Аннотация с использованием `exec_annotation`:**
   - Используйте команду `exec_annotation` для аннотации вашего FASTA файла.
   - Выходные данные сохраните в файл `annotation.tsv`.

Ответ:

Запуск exec_annotation:

exec_annotation -p profile -o annotation.tsv proteins.fasta

Определение пороговых значений:

get_thres -p profile -o threshold.tsv

Фильтрация результата:

filter_annotation -a annotation.tsv -t threshold.tsv -o filtered_annotation.tsv

Критерий оценивания самостоятельной работы – результаты по каждой работе оформляются по указанным требованиям (смотрите в описании задания) и загружаются на образовательный портал ИГУ (https://educa.isu.ru/). Преподаватель оценивает задания, если все решено верно, студен получает зачет по заданию, если имеются недочеты или ошибки, задание отправляется на доработку с указанием допущенных ошибок. Отчёт по переработанному заданию загружается на образовательный портал для повторного оценивания.

3.Оценочные средства для промежуточной аттестации

Промежуточная аттестация проходит в форме зачета (7 семестр), к которому допускаются студенты, выполнившие в полном объеме аудиторную нагрузку, самостоятельную работу. Студенты, имеющие задолженность, должны выполнить все обязательные виды деятельности.

Фонд оценочных средств для промежуточной аттестации включает:

- тестовые задания для зачета.

Назначение оценочных средств: выявить сформированность компетенций ПК-1, ОПК-7 (см. п. III).

Тестовое задание включает два варианта по 20 вопросов по всем темам курса. К тесту допускаются студенты, выполнившие все домашние заданий и получившие по каждому заданию зачет.

Критерий оценивания тестового экзаменационного задания

№	Тип задания	Критерии оценки	Результат оценивания
1	Задание закрытого типа на установление соответствия	Считается верным, если правильно установлены все соответствия (позиции одного столбца верно соотнесены с позициями другого столбца)	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
2	Задание закрытого типа на установление последовательности	Считается верным, если правильно указана вся последовательность цифр	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
3	Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора	Считается верным, если правильно указана цифра (буква) правильного ответа и приведены корректные аргументы, используемые при выборе ответа	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
4	Задание комбинированного типа с выбором нескольких верных ответов из четырех предложенных и обоснованием выбора	Считается верным, если правильно указаны цифры (буквы) правильного ответа и приведены корректные аргументы, используемые при выборе ответа	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов

5	Задание открытого типа с развернутым ответом	Считается верным, если ответ совпадает с эталонным ответом по содержанию и полноте	Полное соответствие эталонному ответу – 1 балл Все остальные случаи – 0 баллов

Система получения баллов за тестирование

Оценка	**критерий**
зачтено	15 и более баллов
не зачтено	14 баллов и менее

3.1 Оценочные материалы для промежуточной аттестации (зачет)

Тестирование (Вариант 1).

Индекс и содержание формируемой компетенции	**Индикаторы компетенций**	**Тестовые задания для промежуточной аттестации**
ПК-1 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	*ИДК ОПК-6.1* Знает принципы создания компьютерных программ, используемых в биоинформатике и биоинженерии *ИДК ОПК-6.2* Использует современные IT-технологии при сборе, анализе, обработке и представлении информации. *ИДК ОПК-6.3* Использует навыки создания компьютерных программ, баз данных и иные программных продуктов, используемых в	**Задание комбинированного типа с выбором одного или нескольких верных ответов из четырех предложенных с аргументацией выбора** ***Вопрос 1.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой метод оценки доверительного интервала чаще всего используется в бутстрэп-методе для одномерных статистик? A) Параметрический метод B) Непараметрический бутстрэп C) Байесовский метод D) Метод моментов Ответ _____ Правильный ответ: B Аргументация: Непараметрический бутстрэп наиболее распространен при оценке доверительных интервалов, так как он не требует предположений о форме распределения. ***Вопрос 2.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* При сравнении средних двух выборок с помощью бутстрэп-метода, какую статистику чаще всего используют? A) Среднее квадратическое отклонение B) Медиану C) Разность средних D) Максимум значений Ответ _____ Правильный ответ: C Аргументация: Разность средних — это стандартная статистика при сравнении двух групп.

	биоинженерии и биоинформатике.	***Вопрос 3.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* В чем преимущество бутстрэп-метода в регрессионном анализе? A) Повышает скорость сходимости B) Избавляет от необходимости в независимости ошибок C) Позволяет оценить устойчивость оценок без строгих предпосылок о распределении D) Заменяет априорные знания Ответ _____ Правильный ответ: C Аргументация: Бутстрэп позволяет получить доверительные интервалы регрессионных коэффициентов, не полагаясь на нормальность остатков. ***Вопрос 4.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что является основной идеей перестановочного теста? A) Формирование теоретического распределения B) Повторный выбор с возвращением C) Использование всех возможных перестановок меток групп D) Минимизация функции потерь Ответ _____ Правильный ответ: C Аргументация: Перестановочные тесты основываются на случайной или полной перестановке меток для оценки распределения статистики при H_0. ***Вопрос 5.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая функция используется в методе максимального правдоподобия? A) Логарифм функции плотности B) Функция правдоподобия C) Функция распределения D) Дисперсионная функция Ответ _____ Правильный ответ: B Аргументация: Метод максимального правдоподобия оптимизирует именно функцию правдоподобия — вероятность наблюдаемых данных при заданных параметрах модели.
ОПК-7 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	*ИДК ОПК-7.1* Демонстрирует теоретические и практические навыки использования современных информационных технологий в области профессианальной деятельности.	
	ИДК ОПК-7.2 Использует современные информационные технологии в рамках освоения материала и реализации задач в области профессианальной деятельности	

		Вопрос 6. *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Цепи Маркова используются в МСМС для: A) Оценки точных аналитических решений B) Перехода между конфигурациями модели с заданным распределением C) Минимизации ошибок модели D) Ускорения градиентного спуска Ответ: B Аргументация: Цепи Маркова моделируют зависимость между состояниями, а МСМС использует это для приближения распределений. ***Вопрос 7.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой из процессов описывается уравнением Колмогорова? A) Детерминированный процесс B) Процесс с постоянной скоростью C) Стохастический процесс переходных вероятностей D) Нелинейный хаос Ответ _____ Правильный ответ: C Аргументация: Уравнение Колмогорова описывает эволюцию переходных вероятностей в стохастических моделях. ***Вопрос 8.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что такое «регулярное выражение» в контексте анализа биологических текстов? A) Формула дисперсии B) Уравнение правдоподобия C) Шаблон для поиска текста по правилам D) Метод оптимизации Ответ _____ Правильный ответ: C Аргументация: Регулярные выражения задают шаблоны для извлечения информации из текстов (например, аннотаций генов).

		Вопрос 9. *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Где наиболее широко применяются скрытые Марковские модели в биоинформатике? A) Моделирование метаболических путей B) Анализ сезонных колебаний C) Распознавание мотивов в последовательностях ДНК D) Построение деревьев родства Ответ: C Аргументация: HMM широко используются для идентификации функциональных элементов в биологических последовательностях. ***Вопрос 10.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что является ключевым преимуществом бутстрэп-метода по сравнению с классическими методами оценки доверительных интервалов? A) Быстрая сходимость B) Универсальность при любом распределении данных C) Не требует выборки D) Используется только при нормальном распределении Ответ _____ Правильный ответ: B Аргументация: Бутстрэп-метод не требует нормальности и применяется при любом виде распределения. ***Вопрос 11.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой R-пакет используется для анализа PERMANOVA? A) dplyr B) vegan C) MASS D) boot Ответ _____ Правильный ответ: B Аргументация: PERMANOVA реализован в пакете vegan, для экологической статистики.

		Вопрос 12. *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что оценивает метод Монте-Карло при использовании в биоинформатике? A) Точное аналитическое решение B) Вероятностные распределения параметров C) Постоянную Планка D) Уровень корреляции Ответ _____ Правильный ответ: B Аргументация: Метод Монте-Карло применяют для оценки распределений параметров при неизвестных теоретических формах. ***Вопрос 13.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Каково основное назначение логарифма правдоподобия? A) Для интерпретации частот B) Упростить математические вычисления C) Минимизировать ошибки D) Стандартизировать данные Ответ _____ Правильный ответ: B Аргументация: Логарифм правдоподобия используется для удобства вычислений — он превращает произведение в сумму. **Задание закрытого типа на установление соответствия** ***Вопрос14.*** *Прочитайте вопрос и установите соответствие.* Каковы назначения следующих функций в языке программирования R? a) sample b) mean c) quantile d) replicate Ответ _____ Правильный ответ: a — 2 (Создание случайной выборки)

		b — 1 (Вычисление среднего) c — 4 (Оценка квантили распределения) d — 3 (Повтор многократных вычислений) ***Вопрос 15.*** *Прочитайте вопрос и установите соответствие.* Для чего используются функции: a) grep b) sub c) gsub d) strsplit Ответ _____ Правильный ответ: a — 1 (Поиск совпадений с регулярным выражением) b — 2 (Замена первого совпадения) c — 3 (Замена всех совпадений) d — 4 (Разделение строки по шаблону) **Задание закрытого типа на установление последовательности** ***Вопрос 16.*** *Прочитайте вопрос и установите последовательность.* Расположите следующие шаги в правильной последовательности, описывающей использование бутстреп-метода для расчета доверительного интервала для среднего значения выборки. Шаги: A. Повторить шаги 2 и 3 большое количество раз (например, 10000 раз), чтобы получить большое количество бутстреп-выборок и их средних значений. B. Рассчитать среднее значение для каждой бутстреп-выборки. C. Исходная выборка данных. D. Выбрать доверительный уровень (например, 95%). E. Оценить доверительный интервал на основе распределения полученных средних значений бутстреп-выборок. Например, для 95% доверительного интервала, взять 2.5-й и 97.5-й процентили полученного распределения средних значений. F. Сгенерировать бутстреп-выборку путем случайного выбора с возвращением n элементов из исходной выборки, где n - размер исходной выборки. Правильный ответ ответ: C - D - F - B - A - E

Вопрос 17.
Прочитайте вопрос и установите последовательность.
Расположите следующие шаги в правильной последовательности, описывающей использование бутстреп-метода для расчета доверительного интервала коэффициента корреляции (например, коэффициента Пирсона) между двумя переменными.
Шаги:
A. Повторить шаги 2 и 3 большое количество раз (например, 10000 раз), чтобы получить большое количество бутстреп-оценок коэффициента корреляции.
B. Рассчитать коэффициент корреляции между двумя переменными для каждой бутстреп-выборки.
C. Исходные данные, состоящие из пар значений двух переменных (например, X и Y).
D. Выбрать доверительный уровень (например, 95%).
E. Оценить доверительный интервал на основе распределения полученных бутстреп-оценок коэффициента корреляции. Например, для 95% доверительного интервала, взять 2.5-й и 97.5-й процентили полученного распределения коэффициентов корреляции.
F. Сгенерировать бутстреп-выборку путем случайного выбора с возвращением n пар значений из исходного набора данных, где n - размер исходного набора данных. Важно сохранить соответствие между значениями X и Y в каждой паре.
Правильный ответ:
C - D - F - B - A - E

Задание открытого типа с развернутым ответом

Вопрос 18.
Прочитайте вопрос и запишите развернутый обоснованный ответ.
Использование регулярных выражений в языке программирования R для анализа биологических текстов примерами
Правильный ответ:
Регулярные выражения (regex) в R - это мощный инструмент для работы с текстовыми данными. Они позволяют находить, извлекать, заменять и проверять соответствие шаблонам в строках. R предоставляет ряд функций для работы с regex, основанных на движке PCRE (Perl Compatible Regular Expressions), обеспечивая широкие возможности для обработки текста.
Основные функции для работы с регулярными выражениями в R:

- grep(pattern, x, ignore.case = FALSE, value = FALSE, ...): Ищет совпадения с шаблоном pattern в векторе строк x.
  - pattern: Регулярное выражение для поиска.
  - x: Вектор строк для поиска.
  - ignore.case = TRUE: Игнорировать регистр при поиске.
  - value = TRUE: Возвращает совпавшие элементы x вместо индексов.
  - Возвращает индексы элементов x, в которых найдены совпадения, или сами элементы, если value = TRUE.
- grepl(pattern, x, ignore.case = FALSE, ...): Аналогичен grep, но возвращает логический вектор, указывающий,

содержит ли каждый элемент x совпадение с pattern.

- sub(pattern, replacement, x, ignore.case = FALSE, ...): Заменяет *первое* совпадение с pattern в каждой строке x на replacement.
  - replacement: Строка, на которую заменяется совпадение.
  - Возвращает вектор строк с выполненными заменами.
- gsub(pattern, replacement, x, ignore.case = FALSE, ...): Заменяет *все* совпадения с pattern в каждой строке x на replacement.
- regexpr(pattern, text, ignore.case = FALSE, perl = TRUE, useBytes = FALSE): Находит позицию первого совпадения с pattern в строке text. Возвращает стартовую позицию совпадения (или -1, если совпадения не найдены) и атрибуты: "match.length" (длина совпадения) и "useBytes" (использовались ли байты).
- gregexpr(pattern, text, ignore.case = FALSE, perl = TRUE, useBytes = FALSE): Находит позиции *всех* совпадений с pattern в строке text. Возвращает список, где каждый элемент соответствует строке text, и содержит вектор стартовых позиций совпадений (или -1, если совпадения не найдены).

Пример использования:

```
# Замена всех цифр на символ '*'
text <- c("Price: $10", "Discount: 20%", "Total: $30")
replaced_text <- gsub("[0-9]", "*", text)
print(replaced_text) # Output: [1] "Price: $**"    "Discount: **%"  "Total: $**"
```

Вопрос 19.

Прочитайте вопрос и запишите развернутый обоснованный ответ.

Принцип метода максимального правдоподобия для анализа типа распределения случайных величин?

Правильный ответ:

Принцип метода максимального правдоподобия (MLE) не используется *непосредственно* для определения типа распределения случайных величин. MLE предполагает, что тип распределения *уже известен* и используется для оценки *параметров* этого распределения. Тем не менее, ММП играет *косвенную* роль в выборе типа распределения. Вот как это работает:

1. Предположение о распределении (несколько вариантов):

Прежде чем применять MLE, необходимо *предположить*, какие возможные распределения могут описывать ваши данные. Это может быть основано на:

- Теоретических знаниях: Например, если вы анализируете время между событиями, экспоненциальное распределение может быть хорошим кандидатом.
- Визуальном анализе данных: Гистограмма или Q-Q plot могут подсказать, какие распределения могут подходить (например, симметричная гистограмма может указывать на нормальное распределение).

Вам придется рассмотреть *несколько* кандидатов (например, нормальное, экспоненциальное, гамма, логнормальное, Вейбулла и т.д.).

2. Применение MLE для каждого предполагаемого распределения:

Для каждого из выбранных распределений выполняются следующие шаги:

		• Формулировка функции правдоподобия: Определяется функция правдоподобия для каждого распределения, выражающая вероятность наблюдения вашей выборки данных, как функцию параметров этого распределения. • Максимизация функции правдоподобия: Находятся значения параметров, которые максимизируют функцию правдоподобия для каждого распределения. Это дает *оценки максимального правдоподобия (MLE)* для параметров каждого распределения. 3. Оценка соответствия модели данным (Model Selection): После применения MLE для каждого предполагаемого распределения, необходимо оценить, насколько хорошо каждое распределение *соответствует* вашим данным. Это делается с помощью различных критериев, *основанных на функции правдоподобия*: • Логарифмическое правдоподобие (Log-Likelihood): Чем выше значение логарифмического правдоподобия, тем лучше модель соответствует данным. Однако, простое сравнение логарифмического правдоподобия может привести к переобучению (выбору более сложной модели, которая лучше соответствует конкретной выборке, но плохо обобщается на новые данные). • Критерий Акаике (Akaike Information Criterion, AIC): AIC учитывает не только логарифмическое правдоподобие, но и количество параметров в модели, штрафуя за излишнюю сложность. Формула: AIC = -2 * log-likelihood + 2 * k, где k - количество параметров в модели. Меньшее значение AIC указывает на лучшее соответствие модели данным, с учетом ее сложности. • Байесовский информационный критерий (Bayesian Information Criterion, BIC) или критерий Шварца (Schwarz Information Criterion, SIC): BIC аналогичен AIC, но более сильно штрафует за сложность модели, особенно при больших размерах выборки. Формула: BIC = -2 * log-likelihood + k * log(n), где n - размер выборки. Меньшее значение BIC указывает на лучшее соответствие модели данным. ***Вопрос 20.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Принцип работы теста PERMANOVA (Permutational Multivariate Analysis of Variance) в анализе многомерных данных? Правильный ответ: PERMANOVA (Permutational Multivariate Analysis of Variance) — это непараметрический статистический тест, используемый для анализа различий между группами в многомерных данных. В отличие от традиционного MANOVA (Multivariate Analysis of Variance), PERMANOVA не требует, чтобы данные соответствовали предположениям о нормальности и гомогенности дисперсий, что делает его более подходящим для анализа экологических и других типов данных, где эти предположения часто нарушаются. Основные идеи и этапы работы PERMANOVA: 1. Многомерные данные: PERMANOVA работает с матрицей данных, где каждая строка представляет собой образец (например, участок леса, пациент), а каждый столбец представляет собой переменную (например, виды растений, уровни экспрессии генов). Таким образом, каждый образец описывается вектором значений по нескольким переменным. 2. Матрица расстояний (Dissimilarity matrix): Первым шагом PERMANOVA является преобразование матрицы данных в матрицу расстояний (также называемую матрицей несходства). Матрица расстояний содержит значения,

		отражающие попарные расстояния (несходства) между всеми образцами. Существуют различные метрики расстояний, которые можно использовать, такие как Евклидово расстояние, расстояние Брея-Куртиса (Bray-Curtis dissimilarity) (особенно популярное в экологии), расстояние Махаланобиса и др. Выбор метрики расстояний зависит от природы данных и исследовательских вопросов. 3. Разбиение общей изменчивости (Partitioning variance): PERMANOVA разделяет общую изменчивость (общую сумму квадратов) в матрице расстояний на компоненты, объясняемые различными факторами (группами, предикторами). Этот процесс аналогичен тому, как ANOVA разделяет изменчивость в одномерных данных. 4. Формулировка гипотез: o Нулевая гипотеза (H0): Не существует значимых различий между группами в многомерном пространстве, т.е. распределения образцов в разных группах идентичны. o Альтернативная гипотеза (H1): Существуют значимые различия между группами в многомерном пространстве, т.е. распределения образцов в разных группах отличаются. 5. Расчет статистики F (Pseudo-F statistic): PERMANOVA вычисляет статистику F (часто называемую "pseudo-F statistic"), которая представляет собой отношение изменчивости между группами к изменчивости внутри групп. Эта статистика измеряет, насколько велика разница между группами по сравнению с разницей внутри групп. 6. Пермутационный тест (Permutation test): В отличие от традиционного ANOVA, который использует F-распределение для определения значимости, PERMANOVA использует пермутационный тест. Пермутационный тест состоит из следующих шагов: o Перемешивание меток групп: Метки групп (которые определяют, к какой группе принадлежит каждый образец) случайным образом перемешиваются между образцами. o Пересчет статистики F: Для каждого перемешивания вычисляется новая статистика F на основе перемешанных меток групп. o Повторение: Процесс перемешивания и пересчета F-статистики повторяется большое количество раз (например, 999, 9999 раз). o Вычисление p-значения: P-значение вычисляется как доля перестановок, для которых F-статистика, полученная на перемешанных данных, больше или равна F-статистике, полученной на исходных данных. То есть, p-значение показывает вероятность получить наблюдаемую или более экстремальную разницу между группами случайно, если нулевая гипотеза верна. 7. Принятие решения: Если p-значение меньше заданного уровня значимости (обычно 0.05), то нулевая гипотеза отклоняется, и делается вывод о том, что существуют значимые различия между группами в многомерном пространстве.

Тестирование (Вариант 2).

Индекс и содержание формируемой	**Индикаторы компетенций**	**Тестовые задания для промежуточной аттестации**

компетенции		
ПК-1 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.	*ИДК ОПК-6.1* Знает принципы создания компьютерных программ, используемых в биоинформатике и биоинженерии.	**Задание комбинированного типа с выбором одного или нескольких верных ответов из четырех предложенных и аргументацией выбора** ***Вопрос 1.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой шаг первым выполняется при реализации базового бутстрэп-алгоритма в R? A) Расчет стандартной ошибки B) Создание бутстрэп-выборок с возвращением C) Визуализация распределения D) Построение регрессионной модели Ответ _____ Правильный ответ: B Аргументация: Бутстрэп начинается с генерации большого количества выборок с возвращением из исходных данных. ***Вопрос 2.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* В чем отличие бутстрэп-метода от классического t-теста при проверке гипотез? A) Требует равенства дисперсий B) Не требует предположений о распределении C) Использует только дискретные данные D) Требует независимости наблюдений Ответ _____ Правильный ответ: B Аргументация: Бутстрэп — непараметрический метод, не требующий нормальности и равенства дисперсий, в отличие от t-теста. ***Вопрос 3.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой тип бутстрэпа используется при коррелированном времени или пространственных данных? A) Параметрический бутстрэп B) Блочный бутстрэп (block bootstrap) C) Простая выборка без возвращения
	ИДК ОПК-6.2 Использует современные IT-технологии при сборе, анализе, обработке и представлении информации.	
	ИДК ОПК-6.3 Использует навыки создания компьютерных программ, баз данных и иные программных продуктов, используемых в биоинженерии и биоинформатике	
ОПК-7 Способен понимать	*ИДК ОПК-7.1* Демонстрирует теоретические и	

принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.	практические навыки использования современных информационных технологий в области профессиональной деятельности. *ИДК ОПК-7.2* Использует современные информационные технологии в рамках освоения материала и реализации задач в области профессиональной деятельности.	D) Джекнайф Ответ _____ Правильный ответ: B Аргументация: При автокоррелированных данных используют блочный бутстрэп, где выборки берутся блоками, чтобы сохранить зависимость. ***Вопрос 4.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая ключевая гипотеза проверяется в перестановочном тесте для двух групп? A) Группы имеют одинаковую медиану B) Группы взяты из одного и того же распределения C) В одной группе больше дисперсия D) Распределения симметричны Ответ _____ Правильный ответ: B Аргументация: Перестановочные тесты проверяют, являются ли группы эквивалентными по распределению. ***Вопрос 5.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что означает значение логарифма функции правдоподобия близкое к нулю? A) Высокая вероятность модели B) Модель точно предсказывает данные C) Данные слабо соответствуют модели D) Параметры модели не влияют на результат Ответ _____ Правильный ответ: C Аргументация: Логарифм функции правдоподобия ближе к нулю — значит, сама вероятность мала: модель плохо описывает данные. ***Вопрос 6.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что делает алгоритм Метрополиса-Гастингса в рамках MCMC? A) Максимизирует функцию потерь B) Строит дерево решений C) Генерирует цепь с заданным стационарным распределением

		D) Рассчитывает p-значения Ответ _____ Правильный ответ: C Аргументация: Метод Метрополиса-Гастингса обеспечивает генерацию выборок из сложного распределения с помощью цепи Маркова. ***Вопрос 7.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая задача может быть решена с помощью модели случайных блужданий (random walk)? A) Расчет вероятности мутации в одной позиции гена B) Построение филогенетического дерева C) Моделирование миграции особей в пространстве D) Выделение генов из аннотаций Ответ _____ Правильный ответ: C Аргументация: Случайное блуждание используется в моделях перемещения особей или диффузии в биологических системах. ***Вопрос 8.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой символ регулярных выражений в R соответствует "любому символу"? A) ^ B) * C) . D) $ Ответ: C Аргументация: В регулярных выражениях . означает "любой одиночный символ", это базовая конструкция шаблона. ***Вопрос 9.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какова основная структура скрытой Марковской модели? A) Последовательность регрессионных уравнений B) Набор скрытых состояний и наблюдаемых выходов C) Упорядоченное дерево D) Матрица корреляции

		Ответ _____ Правильный ответ: B Аргументация: HMM состоит из скрытых (невидимых) состояний, переходов между ними и вероятностей наблюдаемых выходов. ***Вопрос 10.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Сколько бутстрэп-репликаций обычно достаточно для устойчивой оценки параметра? A) 5 B) 25 C) 100–200 D) 1000 и более Ответ _____ Правильный ответ: D Аргументация: Для надежных доверительных интервалов обычно используют от 1000 бутстрэп-репликаций и выше. ***Вопрос 11.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой тип модели можно протестировать с помощью PERMANOVA? A) Линейную модель с независимыми остатками B) Модель сходства между группами по множественным переменным C) Временные ряды D) Иерархическую кластеризацию Ответ _____ Правильный ответ: B Аргументация: PERMANOVA оценивает различия между группами в многомерном пространстве (например, по видам), используя матрицу расстояний. ***Вопрос 12.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* В чем заключается преимущество MCMC при выборе регрессионных моделей? A) Всегда приводит к линейной модели B) Избегает проблемы мультиколлинеарности C) Позволяет оценить распределение параметров без строгих предпосылок

		D) Повышает точность метода наименьших квадратов Ответ _____ Правильный ответ: C Аргументация: MCMC позволяет исследовать апостериорные распределения параметров без строгих предположений о распределениях ошибок. ***Вопрос13.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Для чего используется команда grep() в языке R? A) Вычисление вероятности B) Построение бутстрэп-интервала C) Поиск совпадений по регулярному выражению D) Математическая оптимизация Ответ: C Аргументация: grep() используется для поиска строк, соответствующих регулярному выражению, и широко применяется при анализе текстов. **Задание закрытого типа на установление соответствия** ***Вопрос 14.*** *Прочитайте вопрос и установите соответствие.* Каковы функции в контексте анализа PERMANOVA в R? Функции: a) adonis b) vegdist c) permute d) set.seed Анализ PERMANOVA в R: 1 (Проведение PERMANOVA) 2 (Построение матрицы расстояний) 3 (Настройка схемы перестановок) 4 (Фиксация генератора случайных чисел) Ответ _____ Правильный ответ: a — 1 (Проведение PERMANOVA) b — 2 (Построение матрицы расстояний)

		c — 3 (Настройка схемы перестановок) d — 4 (Фиксация генератора случайных чисел) ***Вопрос 15.*** *Прочитайте вопрос и установите соответствие.* Каковы назначения следующих компонентов в анализе HMM? Компоненты: a) Viterbi b) BaumWelch c) emission d) hmm Анализ HMM: 1 (Создание скрытой Марковской модели) 2 (Поиск самой вероятной последовательности скрытых состояний) 3 (Оценка параметров модели) 4 (Определение вероятностей наблюдаемых состояний) Ответ _____ Правильный ответ: a — 2 (Поиск самой вероятной последовательности скрытых состояний) b — 3 (Оценка параметров модели) c — 4 (Определение вероятностей наблюдаемых состояний) d — 1 (Создание скрытой Марковской модели) **Задание закрытого типа на установление последовательности** ***Вопрос 16.*** *Прочитайте вопрос и установите последовательность.* Расположите следующие шаги в правильной последовательности, описывающей использование бутстреп-метода для расчета доверительного интервала для стандартного отклонения выборки. **Шаги:** A. Повторить шаги 2 и 3 большое количество раз (например, 10000 раз), чтобы получить большое количество бутстреп-выборок и их стандартных отклонений. B. Рассчитать стандартное отклонение для каждой бутстреп-выборки. C. Исходная выборка данных. D. Выбрать доверительный уровень (например, 95%). E. Оценить доверительный интервал на основе распределения полученных стандартных отклонений бутстреп-выборок. Например, для 95% доверительного интервала, взять 2.5-й и 97.5-й процентили полученного распределения

		стандартных отклонений. F. Сгенерировать бутстреп-выборку путем случайного выбора с возвращением n элементов из исходной выборки, где n - размер исходной выборки. Правильный ответ: C - D - F - B - A - E ***Вопрос 17.*** *Прочитайте вопрос и установите последовательность.* Расположите следующие шаги в правильной последовательности, описывающей использование бутстреп-метода для оценки p-значения, связанного с разницей средних значений в двух независимых выборках. Это поможет определить, является ли разница средних статистически значимой. A. Рассчитать наблюдаемую разницу средних значений между двумя исходными выборками. B. Сгенерировать бутстреп-выборки для каждой группы путем случайного выбора с возвращением n1 и n2 элементов из перемешанных данных, где n1 и n2 - размеры исходных выборок. C. Перемешать (пул) обе выборки вместе, чтобы создать единую совокупность данных. Это делается в предположении, что нулевая гипотеза (отсутствие различий) верна. D. Выбрать количество бутстреп-репликаций (например, 10000). E. Рассчитать разницу средних значений для каждой пары бутстреп-выборок. F. Рассчитать p-значение как долю бутстреп-разностей средних, которые имеют абсолютное значение, большее или равное наблюдаемой разнице средних (шаг A). G. Повторить шаги B и E выбранное количество раз (шаг D). H. Исходные данные: две независимые выборки (например, выборка X и выборка Y). Правильный ответ: H - A - D - C - B - G - E – F **Задание открытого типа с развернутым ответом** ***Вопрос 18.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Принцип работы перестановочного теста для тестирования статистических гипотез? Правильный ответ: Перестановочный тест (также известный как рандомизационный тест или точный тест) - это непараметрический статистический тест, используемый для проверки гипотез о различиях между группами или о связи между переменными. Он является мощной альтернативой параметрическим тестам, особенно когда предположения о нормальности распределения или равенстве дисперсий нарушаются. Основные идеи и этапы работы перестановочного теста: 1. Формулировка гипотез: o Нулевая гипотеза (H0): Не существует связи между переменными или различий между группами. В контексте

		сравнения групп, это означает, что группы происходят из одного и того же распределения. o Альтернативная гипотеза (H1): Существует связь между переменными или различия между группами. Альтернативная гипотеза может быть односторонней (например, группа A больше группы B) или двусторонней (группы A и B различаются). 2. Выбор тестовой статистики (Test Statistic): Выбирается тестовая статистика, которая отражает разницу, которую мы хотим обнаружить. Примеры: o Разница средних: Для сравнения двух групп по количественной переменной. o Разница медиан: Для сравнения двух групп по количественной переменной (более устойчива к выбросам, чем разница средних). o Коэффициент корреляции: Для проверки связи между двумя количественными переменными. o Статистика хи-квадрат: Для проверки связи между двумя категориальными переменными. o Выбор тестовой статистики должен соответствовать типу данных и гипотезе. 3. Вычисление наблюдаемой тестовой статистики: Вычисляется значение выбранной тестовой статистики на основе *исходных* данных. Это значение будет служить отправной точкой для сравнения с результатами перестановок. 4. Генерация перестановок (Permutations): Этот шаг является ключевым в перестановочном тесте. Создается большое количество перестановок данных путем перераспределения значений между группами (или между переменными, в зависимости от задачи) *случайным* образом. o Для сравнения групп: Метки групп (например, "контроль" и "эксперимент") случайным образом переназначаются образцам данных. То есть, берется набор данных и случайным образом перетасовываются значения группы к которой принадлежит каждый элемент. o Для проверки связи между переменными: Значения одной переменной случайным образом перемешиваются относительно значений другой переменной. Важно: Все перестановки должны быть равновероятными при условии верности нулевой гипотезы. 5. Вычисление тестовой статистики для каждой перестановки: Для каждой сгенерированной перестановки данных вычисляется значение выбранной тестовой статистики. Это создает *распределение перестановок* тестовой статистики. 6. Вычисление p-значения: P-значение вычисляется как доля перестановок, для которых значение тестовой статистики, вычисленное на перестановленных данных, *более экстремальное*, чем значение тестовой статистики, вычисленное на исходных данных. "Более экстремальное" определяется направлением альтернативной гипотезы: o Односторонняя альтернативная гипотеза (например, группа A больше группы B): P-значение - это доля перестановок, для которых тестовая статистика больше или равна наблюдаемой тестовой статистике. o Двусторонняя альтернативная гипотеза (группы A и B различаются): P-значение - это доля перестановок, для которых абсолютное значение тестовой статистики больше или равно абсолютному значению наблюдаемой тестовой статистики. В этом случае мы проверяем, насколько далеко отклоняются значения тестовой статистики от нуля в обе стороны.

		7. Принятие решения: Если p-значение меньше заданного уровня значимости (обычно 0.05), то нулевая гипотеза отклоняется, и делается вывод о том, что существует значимая связь между переменными или различия между группами. В противном случае, нет достаточных оснований для отклонения нулевой гипотезы. Преимущества перестановочного теста: • Непараметричность: Не требует предположений о нормальности распределения или равенстве дисперсий. • Точность: Для небольших выборок перестановочный тест может быть более точным, чем параметрические тесты, поскольку он основан на всех возможных перестановках данных (или на достаточно большом их подмножестве). • Гибкость: Может использоваться с различными тестовыми статистиками и для проверки различных типов гипотез. • Интуитивная интерпретация: P-значение имеет простую и понятную интерпретацию: вероятность получить наблюдаемый или более экстремальный результат случайно, если нулевая гипотеза верна. ***Вопрос 19.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Принцип работы алгоритма скрытых Марковских моделей для идентификации функций белков по их аминокислотным последовательностям? Правильный ответ: Скрытые Марковские модели (HMM) - это мощный вероятностный инструмент, широко используемый в биоинформатике для анализа биологических последовательностей, включая аминокислотные последовательности белков. Их применение для предсказания функций белков основано на том, что белки, выполняющие схожие функции, часто имеют схожие аминокислотные последовательности и паттерны, хотя и не всегда очевидные на первый взгляд. HMM позволяют выявлять эти скрытые паттерны и строить модели, которые отражают эволюционные взаимосвязи и структурные особенности белковых семейств. Основные концепции HMM: • Состояния (States): Представляют собой абстрактные "скрытые" состояния, которые не наблюдаются напрямую. В контексте белковых последовательностей, состояния могут соответствовать консервативным регионам, структурным элементам или функциональным доменам белка. • Наблюдения (Observations): Это то, что мы видим непосредственно - в данном случае, аминокислотные остатки (A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V) в белковой последовательности. • Переходы (Transitions): Представляют собой вероятности перехода из одного состояния в другое. Они отражают, насколько вероятно изменение консервативности или структурного элемента в последовательности. • Выбросы (Emissions): Представляют собой вероятности генерации определенного аминокислотного остатка из каждого состояния. Они отражают, какие аминокислоты наиболее часто встречаются в каждом состоянии. Как HMM используются для идентификации функций белков: 1. Обучение HMM (Training): o Набор данных: Собирается набор аминокислотных последовательностей белков, которые уже известны и

		имеют общую функцию (например, семейство ферментов). Это обучающий набор данных. o Построение профиля HMM: На основе обучающего набора строится профиль HMM, который отражает статистические особенности этого белкового семейства. Это включает в себя: ▪ Определение количества состояний: Часто количество состояний выбирается эвристически, основываясь на ожидаемой сложности белкового семейства. Каждое состояние может соответствовать определенному консервативному региону или структурному мотиву. ▪ Оценка вероятностей переходов: Рассчитываются вероятности перехода между состояниями на основе анализа обучающих последовательностей. Эти вероятности отражают, насколько вероятно последовательность "переходит" от одного консервативного региона к другому. ▪ Оценка вероятностей выбросов: Рассчитываются вероятности выброса каждого аминокислотного остатка из каждого состояния на основе обучающих последовательностей. Эти вероятности отражают, какие аминокислоты наиболее часто встречаются в каждом состоянии. o Алгоритм Баума-Велша (Baum-Welch algorithm): Используется итеративный алгоритм Баума-Велша (также известный как алгоритм прямо-обратного распространения) для оптимизации параметров HMM (вероятностей переходов и выбросов) на основе обучающего набора данных. Алгоритм стремится найти параметры HMM, которые максимизируют вероятность наблюдаемых последовательностей в обучающем наборе. 2. Выравнивание последовательности (Sequence Alignment): o После того, как HMM обучена на белковом семействе, она может быть использована для выравнивания новых, неизвестных белковых последовательностей с этим профилем HMM. Это позволяет идентифицировать потенциальные члены этого белкового семейства. o Алгоритм Витерби (Viterbi algorithm): Используется алгоритм Витерби для поиска наиболее вероятной последовательности состояний (пути) через HMM для данной аминокислотной последовательности. Этот путь определяет, какие состояния HMM лучше всего соответствуют различным регионам последовательности. o Выравнивание профиля HMM: Алгоритм Витерби позволяет выровнять последовательность с профилем HMM, определяя соответствие между аминокислотными остатками и состояниями HMM. Это обеспечивает более чувствительное и точное выравнивание, чем традиционные методы, такие как попарное выравнивание. 3. Скоринг (Scoring): o Вероятность соответствия: HMM вычисляет вероятность того, что данная последовательность соответствует профилю HMM. Эта вероятность отражает, насколько хорошо последовательность "вписывается" в модель белкового семейства. o E-value (ожидаемое значение): Часто используется E-value, который представляет собой ожидаемое количество случайных последовательностей, которые получат оценку, по крайней мере, такую же хорошую, как и данная последовательность. Чем меньше E-value, тем более вероятно, что последовательность является истинным членом белкового семейства. 4. Предсказание функции (Function Prediction): o Пороговое значение: Если вероятность соответствия или E-value превышают определенное пороговое

значение, то белок считается членом данного белкового семейства, и ему приписывается функция, связанная с этим семейством.

- Базы данных HMM: Существуют большие базы данных профилей HMM, такие как Pfam и InterPro, которые содержат профили HMM для тысяч белковых семейств и доменов. Эти базы данных можно использовать для быстрого и автоматического предсказания функций новых белков.

Вопрос 20.

Прочитайте вопрос и запишите развернутый обоснованный ответ.

Принцип работы метода максимального правдоподобия для оценки параметров распределений?

Правильный ответ:

Метод максимального правдоподобия (MLE) - это статистический метод, используемый для оценки параметров вероятностного распределения на основе наблюдаемых данных. Основная идея заключается в том, чтобы найти такие значения параметров распределения, которые максимизируют вероятность (правдоподобие) получения наблюдаемого набора данных. Другими словами, мы ищем параметры, при которых наиболее вероятно увидеть именно те данные, которые мы имеем.

Основные этапы работы MLE:

1. **Выбор распределения:**
   - Необходимо выбрать вероятностное распределение, которое, по вашему мнению, наилучшим образом описывает данные. Например, если данные представляют собой количество успехов в серии независимых испытаний, то можно использовать биномиальное распределение. Если данные непрерывные и выглядят симметрично, можно использовать нормальное распределение.
   - Выбор распределения является важным шагом, так как он определяет форму функционала правдоподобия и, следовательно, результаты оценки.
2. **Запись функции правдоподобия (Likelihood Function):**
   - Функция правдоподобия, обозначаемая как $L(\theta|x)$, где:
     - θ - вектор параметров распределения, которые мы хотим оценить (например, μ и σ для нормального распределения).
     - x - наблюдаемый набор данных ($x_1, x_2, \ldots, x_n$).
   - Функция правдоподобия представляет собой совместную плотность вероятности (или функцию вероятности для дискретных распределений) наблюдаемых данных, рассматриваемую как функция от параметров θ, при фиксированных данных x.
   - Если данные независимы и одинаково распределены (i.i.d.), то функция правдоподобия равна произведению плотностей вероятности (или функций вероятности) для каждого наблюдения:
   - $L(\theta|x) = f(x_1; \theta) * f(x_2; \theta) * \ldots * f(x_n; \theta)$
     где $f(x_i; \theta)$ - плотность вероятности (или функция вероятности) i-го наблюдения x_i при заданных параметрах θ.
3. **Логарифмирование функции правдоподобия (Log-Likelihood Function):**
   - Вместо максимизации функции правдоподобия часто удобнее максимизировать ее логарифм, называемый

		логарифмической функцией правдоподобия (log-likelihood function), обозначаемой как $l(\theta\|x)$ или $\ln(L(\theta\|x))$. o Логарифмирование упрощает вычисления, особенно когда функция правдоподобия является произведением многих множителей. Логарифм произведения превращается в сумму логарифмов: o $l(\theta\|x) = \ln(L(\theta\|x)) = \ln(f(x_1; \theta)) + \ln(f(x_2; \theta)) + ... + \ln(f(x_n; \theta))$ o Максимизация функции правдоподобия эквивалентна максимизации логарифмической функции правдоподобия, поскольку логарифм - монотонно возрастающая функция. 4. **Поиск максимума логарифмической функции правдоподобия:** o Цель - найти значения параметров θ, которые максимизируют $l(\theta\|x)$. Это можно сделать аналитически или численно. o **Аналитический метод:** ▪ Находим производные логарифмической функции правдоподобия по каждому параметру: $\partial l(\theta\|x)/\partial\theta_i$. ▪ Приравниваем производные к нулю и решаем полученную систему уравнений относительно параметров θ. Решения этой системы уравнений называются оценками максимального правдоподобия (MLE). ▪ Проверяем, что найденные решения соответствуют максимуму (а не минимуму или седловой точке) с помощью второй производной или других методов. o **Численные методы:** ▪ Если аналитическое решение недоступно (что часто бывает), используются численные методы оптимизации, такие как градиентный спуск, метод Ньютона-Рафсона или другие алгоритмы. ▪ Численные методы ищут максимум логарифмической функции правдоподобия итеративно. 5. **Получение оценок параметров:** o Значения параметров θ, которые максимизируют логарифмическую функцию правдоподобия, являются оценками максимального правдоподобия (MLE). o Эти оценки обозначаются как $\hat{\theta}$ (θ с крышкой).

Разработчик:

_____ доцент Букин Ю.С.

(подпись)