

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ по
дисциплине
«БИОИНФОРМАТИКА»

Направление подготовки/специальность	44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль)/специализация	География и биология
Год начала обучения	2026
Форма обучения	очная

Разработано
Заведующий кафедрой вычислительной
математики и кибернетики Бабенко М.Г.

Ставрополь 2026 г.

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1: ВВЕДЕНИЕ В БИОИНФОРМАТИКУ	5
Выполнение работы	5
Контрольные вопросы	10
ЛАБОРАТОРНАЯ РАБОТА № 2: ПОСЛЕДОВАТЕЛЬНОСТИ	12
Выполнение работы	12
Контрольные вопросы	17
ЛАБОРАТОРНАЯ РАБОТА № 3 «ПАРНОЕ ВЫРАВНИВАНИЕ»	19
Выполнение работы	19
Контрольные вопросы	24
Приложения к лабораторной работе № 3.....	25
ЛАБОРАТОРНАЯ РАБОТА № 4 «МНОЖЕСТВЕННОЕ ВЫРАВНИВАНИЕ»	29
Выполнение работы	29
Контрольные вопросы	40
ЛАБОРАТОРНАЯ РАБОТА № 5 «ИЗУЧЕНИЕ ШАБЛОНОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ2	41
Выполнение работы	41
Контрольные вопросы	47
Приложение к лабораторной работе № 5.....	48
ЛАБОРАТОРНАЯ РАБОТА № 6 ФИЛОГЕНЕТИЧЕСКИЕ ДЕРЕВЬЯ	49
Контрольные вопросы	53
Приложения к лабораторной работе № 6:.....	54
ЛАБОРАТОРНАЯ РАБОТА № 7 «ТЕКСТОВЫЕ МЕТОДЫ В БИОИНФОРМАТИКЕ»	57
Теоретические выкладки.....	57
Выполнение работы	58
ЛАБОРАТОРНАЯ РАБОТА № 8 ФИЛОГЕНЕТИЧЕСКИЙ АНАЛИЗ	67
Выполнение работы	67
Контрольные вопросы	67
ЛАБОРАТОРНАЯ РАБОТА №9 «ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ НА ЯЗЫКЕ PYTHON»	69
Задания для самостоятельного выполнения	70
ЛАБОРАТОРНАЯ РАБОТА №10 «ОСНОВЫ РАБОТЫ С PANDAS»	75
Методические указания	75
Задания к выполнению	75
Контрольные вопросы	76
Темы для самостоятельного изучения.....	76
ЛАБОРАТОРНАЯ РАБОТА №11 ПРОСТОЙ АНАЛИЗ ДАННЫХ	78
Задания для выполнения	78
Дополнительные задания.....	78
Контрольные вопросы	79
ЛАБОРАТОРНАЯ РАБОТА №12 «ВИЗУАЛИЗАЦИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕК MATPLOTLIB И SEABORN»	80
Цель работы	80
Задания для выполнения:.....	80
Методические указания	81
Дополнительные задания:.....	81
Контрольные вопросы:	81
ЛАБОРАТОРНАЯ РАБОТА №13 «ПАРНАЯ ЛИНЕЙНАЯ РЕГРЕССИЯ»	83
Содержание работы	83
Методические указания	83
Задания для самостоятельного выполнения	93
Контрольные вопросы	94

Дополнительные задания:.....	94
ЛАБОРАТОРНАЯ №14: МНОЖЕСТВЕННАЯ ЛИНЕЙНАЯ РЕГРЕССИЯ НА ЧИСТЫХ ДАННЫХ.....	95
Содержание работы	95
Методические указания	95
Задания для самостоятельного выполнения	102
Контрольные вопросы	102
Дополнительные задания.....	103
ЛАБОРАТОРНАЯ №15: МЕТОДЫ МЕТРИЧЕСКОЙ КЛАССИФИКАЦИИ	104
Задания для выполнения	104
Методические указания	104
Контрольные вопросы	109
Дополнительные задания.....	109
ЛАБОРАТОРНАЯ №16: КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ.....	111
Задания для выполнения	111
Методические указания	111
Контрольные вопросы	113
Дополнительные задания.....	113
ЛАБОРАТОРНАЯ №17: КЛАССИФИКАЦИЯ ТЕКСТОВ	115
Задания для выполнения	115
Методические указания	115
Контрольные вопросы	117
Дополнительные задания.....	117
ЛАБОРАТОРНАЯ РАБОТА №18: «ОЦЕНКА ЭФФЕКТИВНОСТИ МОДЕЛЕЙ БИНАРНОЙ КЛАССИФИКАЦИИ».....	119
Содержание работы	119
Методические указания	120
Задания для самостоятельного выполнения	122
Контрольные вопросы	122
ЛАБОРАТОРНАЯ РАБОТА №19 ДИАГНОСТИКА И КРИВЫЕ ОБУЧЕНИЯ.....	124
Задания для выполнения	124
Методические указания	125
Контрольные вопросы	128
Дополнительные задания.....	129
ЛАБОРАТОРНАЯ РАБОТА №20 ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ.....	130
Задания для выполнения	130
Методические указания	131
Контрольные вопросы	171
Дополнительные задания.....	171
ЛАБОРАТОРНАЯ РАБОТА №21: «ОБРАБОТКА ДАННЫХ ДЛЯ МАШИННОГО ОБУЧЕНИЯ».....	172
Задания для выполнения	172
Контрольные вопросы	173
Дополнительные задания.....	173
ЛАБОРАТОРНАЯ №22: КЛАСТЕРИЗАЦИЯ	175
Задания для выполнения	175
Методические указания	175
ЛАБОРАТОРНАЯ №23: ПОНИЖЕНИЕ РАЗМЕРНОСТИ.....	178
Задания для выполнения	178
Методические указания	178

ЛАБОРАТОРНАЯ РАБОТА № 1: ВВЕДЕНИЕ В БИОИНФОРМАТИКУ

Целью этой работы является познакомить студента с двумя важнейшими базами данных в Интернете - GenBank (<http://www.ncbi.nlm.nih.gov/genbank/index.html>) , база данных генов и геномов и Protein Data Bank (PDB) (<http://www.rcsb.org/pdb/>) - база данных структур биологических макромолекул.

Во время выполнения этой работы студент должен узнать и освоить:

1. Поиск нуклеотидной последовательности по названию определенного белка;
2. Поиск научных статей о конкретном белке;
3. Получать файл, описывающий 3D структуру белка;
4. Изучение третичной структуры белка с использованием 3D-браузера.

Выполнение работы

ENTREZ машина на NCBI

US National Centre for Biotechnological Information (NCBI <http://www.ncbi.nlm.nih.gov/>) поддерживает базу данных GenBank для генов и геномных последовательностей. Этот веб-сервер связывает их с другими базами данных и ресурсами, включая National Library of Medicine <http://www.nlm.nih.gov/> и предоставляет пакет программ для поиска по GenBank - BLAST <http://blast.ncbi.nlm.nih.gov/Blast.cgi>. Этот сервер сделан и поддерживается профессионально. Доступ к нему бесплатный и штат отвечает даже на технические вопросы. Это - лучшее место в Интернете для начала поиска биоинформатических данных.

а) В веб браузере откройте url <http://www.ncbi.nlm.nih.gov/Entrez> . Несмотря на то, что мы будем использовать только один раздел на сервере (и сможем иметь доступ к нему напрямую) стоит потратить несколько минут для того, чтобы посмотреть домашнюю страницу сайта для того, чтобы оценить его масштаб.

б) Кликните по ссылке Entrez. <http://www.ncbi.nlm.nih.gov/Entrez/> Вы

перейдете на обновленную страницу Entrez (по французски «enter», произносится как “awntray”). На этой странице можно производить поиск по нескольким базам данных одновременно. Для этого упражнения нас интересует фермент isocitrate dehydrogenase, играющий важную роль в генерировании энергии. Мы будем искать ген, кодирующий этот белок, в кишечной бактерии, *Escherichia coli*. Мы будем сравнивать бактериальный белок и соответствующий белок млекопитающих.

в) Наберите isocitrate dehydrogenase в поле поиска и нажмите Go. Результатом будут соответствующие записи из нескольких баз данных. Нас интересуют данные из баз Nucleotide (DNA и RNA), Genome

(целые хромосомы организмов) и Structure (трехмерные структуры белков).

Получение нуклеотидной последовательности

а) Кликните на Nucleotide: вы увидите первую из сотни страниц результатов. Нам нужно как-то сократить этот список до размеров, когда с ним будет удобно работать. Чтобы это сделать:

б) Кликните на ссылке Limits. Вы переместитесь на страницу, на которой можно выбрать среди множества опций для ограничения вашего поиска.

в) Выберите Title из выпадающего меню Search field tags и нажмите Search. Вы вернетесь на страницу с результатами и увидите, что общее их количество значительно сократилось. Однако мы хотим ограничить наш поиск последовательностей только *Escherichia coli* (*E. coli*). Поэтому:

г) Добавьте «and E coli» в поле поиска после "isocitrate dehydrogenase". д)

Кликните Search. Вы вернетесь на страницу поиска с двумя результатами, второй из которых - то, что нам нужно.

е) Кликните на номере доступа - J02799. Вы перейдете на страницу в стандартном формате GenBank с документацией и ссылками и следующей за ней последовательностью ДНК. Мы ещё вернемся к этим ссылкам, но вначале для того, чтобы убедиться, что файл может использоваться в программах, нам надо убедиться в том, что он содержит правильный текст.

ж) Выберите опцию Text в выпадающем меню, начинающемся с Send To. Это генерирует страницу в формате plain text без ссылок.

з) Выберите Save из меню File браузера и сохраните страницу под именем J02799.gbк. Тонкость состоит в том, что, хотя этот файл содержит только ASCII текст, он в формате Unix. ASCII символ, обозначающий конец строки, различен в Unix, PC и Mac- платформах.

и) Первая строка показывает нам, что длина последовательности 1568 bp (пар оснований) и последовательность есть ДНК. Собственно последовательность начинается со строки ORIGIN. Эту строку можно использовать как индикатор при открытии любых файлов GenBank.

к) Если вы посмотрите на строки с отступами, следующими за FEATURES, вы увидите CDS. Это - расположение кодирующей последовательности и содержит часть гена, транслируемой в белок -нуклеотиды 291-1541. Почему не вся генетическая информация транслируется в аминокислоты белка будет объяснено на лекциях. Это имеет практическую ценность для анализа последовательности ДНК.

Получение научных статей

Практикующим ученым часто нужны работы, связанные с последовательностями, которые они скачали из баз данных. NCBI сервер предоставляет ссылки на аннотации статей в National Library of Medicine (MLM) и часто - pdf версии оригиналов. NLM, разумеется, содержит аннотации статей, описывающих не только последовательности ДНК и белков. Все они доступны через машину PubMed из NCBI.

Вернитесь на страницу последовательности с html ссылками (кнопка Back браузера).

а) Кликните на PUBMED номере (3112144). Это перенесет вас на страницу со ссылкой на оригинальную статью, откуда вы можете попасть на веб сайт журнала, откуда статья может быть скачана в виде Pdf файла.

Получение структурной информации о белке

Файл GenBank J02799 представляет белок isocitrate dehydrogenase

(ICDH) как строку символов. Однако белки - это биологические последовательности, и будет очень полезно исследовать этот аспект. 3D структуры белков определяются экспериментально и данные о них депонируются в виде текстовых файлов, описывающих xyz координаты атомов в пространстве. База данных, хранящая эти файлы - Protein Data Bank (бывшая Brookhaven) и формат данных, обычно распознаваемый большинством программ для молекулярной графики называется PDB форматом. GenBank содержит некоторое количество данных из PDB в своей MMDB ((Molecular Modelling Data Base), но также имеет ссылки на Protein Data Bank.

Вернитесь на главную страницу Entrez, где может быть необходимым повторить поиск по многим базам данных по запросу isocitrate dehydrogenase. а)

Кликните на иконке Structure. Это откроет перед вами две страницы ICDH структур. Нам нужно закрыть одну - ту, которая содержит белок со связанным изоцитратом.

б) На второй странице кликните на 5ICD. Вы попадете на страницу этой структуры на MMDB. Используйте RasMol Chime плагин - вы сможете увидеть этот белок. Нам нужно скачать этот файл.

в) Кликните на PDB: 5ICD. Это переместит нас на страницу с этой структурой на сайте PDB <http://www.rcsb.org/pdb/>.

г) Кликните Download File. Вам нужно ввести PDB ID “5ICD” в главном окне. В опциях сохранения выберите “PDB format” и “No compression”. Сохраните файл на диск как 5ICD.pdb.

Использование RasMol для просмотра и анализа структуры белка

Как уже не раз отмечалось, файлы в формате GenBank представляют белки как строки символов 20-и буквенного алфавита. Хотя эти строки и

несут информацию, как, например, ДНК, эта информация реализуется через трехмерную структуру белка и присущие ему свойства. Мы будем использовать бесплатную кросс-платформенную программу RasMol для визуализации этой структуры.

Запустите RasMol, напечатав rasmol в командной строке unix. Для Windows нужно скачать пакет RasMol с <http://www.umass.edu/microbio/rasmol/> или <http://rasmol.org/> и установить его. Вы увидите черный графический экран. RasMol - это графическое приложение с GUI. Но полная его мощь раскрывается через использование командной строки и команд, вводимых в текстовом окне, которое обычно скрыто.

Посмотрите на RasMol Reference Card
[http://www.openrasmol.org/software/RasMol 2.7.3/doc/README.html](http://www.openrasmol.org/software/RasMol%202.7.3/doc/README.html)

(Имеется также on-line manual, содержащий полный набор команд и ключей программы). Загрузить RasMol на локальный компьютер можно также с <http://www.bernstein-plus-sons.com/software/rasmol/> и <http://rasmol.org/>

а) Разверните окно командной строки.

б) Если вы хотите, чтобы фон окна командной строки стал белым, введите "set background white" в этом окне.

в) Загрузите файл 5ICD.pdb при помощи команды Open из меню File.

г) Белок появится в виде wire-frame. Это бывает полезно для биологов, поскольку позволяет изучать отдельные аминокислоты, но в этом случае может быть сложно связать их с первичной структурой белка, то есть с его последовательностью.

д) Введите "restrict protein"

е) Выберите подменю Backbone из меню Display. Вы увидите начало белка, однако его конец будет трудно найти на экране.

ж) Выберите Group (или Индекс) в меню Цвет. Теперь вы можете проследить всю цепь благодаря градиенту цвета от голубого до красного, вращая молекулу.

з) Выберите Spacefill (Молекулярная поверхность) из меню Display. Теперь вы можете наблюдать белок как целостный объект, а не набор точек и линий с промежутками между ними. Но такой вариант просмотра затрудняет изучение внутренних структур.

и) Вы имеете возможность вращать (rotate) молекулу белка при помощи

левой кнопки мыши или правой кнопки мыши + shift key; перемещать (move) молекулу при помощи правой кнопки мыши; увеличивать или уменьшать масштаб изображения при помощи shift key + левой кнопки мыши.

Теперь давайте взглянем на isocitrate, лиганд, связанный с белком. а)

Выберите Wireframe в меню Display.

б) Выберите CPK в меню Colours.

в) В командной строке наберите «select ligand». г)

Выберите Spacefill в меню Display.

Теперь вы видите isocitrate. Очевидно, что он находится внутри молекулы белка.

а) В командной строке наберите «colour blue». б) В

командной строке наберите «select protein». в) Выбе-

рите Spacefill в меню Display.

Вращая молекулу, вы можете увидеть, что молекула, окрашенная в синий цвет (isocitrate) находится в «кармане» на поверхности белка. Так происходит каталитическая реакция.

Необходимо отметить, что изменения в молекуле белка в области связывания изоцитрата вследствие мутации в ДНК может привести к нарушению взаимодействия молекул. Вероятно, что связывания не произойдёт вообще. Однако изменения в других частях молекулы белка могут не повлиять на связывание.

Контрольные вопросы

1. Парное выравнивание. Виды, авторы алгоритмов, цели, значение. Глобальное выравнивание.

2. Вторичные структуры белков, их характеристики и предсказание. ПО и сервисы.

3. Локальное выравнивание. Цели, значение. Алгоритм локального выравнивания.

4. Биоинформатика. Объекты биоинформатики. Задачи, решаемые этой наукой. Методы биоинформатики.

5. Матрицы сравнения последовательностей. PAM, BLOSUM.

6. Почему динамическое программирование используется для парного выравнивания и практически не применяется для выравнивания нескольких последовательностей (MSA)?

ЛАБОРАТОРНАЯ РАБОТА № 2: ПОСЛЕДОВАТЕЛЬНОСТИ

Цель работы:

1. Ознакомиться с пакетом программ EMBOSS;
2. Понять основные принципы декодирования последовательности ДНК в последовательность белка;
3. Понять принципы работы алгоритмов сравнения последовательностей;
4. Получить представление об открытой рамке считывания;
5. Освоить некоторые техники сравнения последовательностей и интерпретации результатов сравнения.

Выполнение работы

EMBOSS <http://emboss.sourceforge.net/> - это "The European Molecular Biology Open Software Suite". EMBOSS является программным пакетом с открытым кодом. Он был специально разработан для нужд молекулярных биологов. Пакет EMBOSS автоматически распознаёт множество биологических форматов данных и позволяет прозрачно для пользователя получать эти данные из сетевых баз. EMBOSS включает в себя большое количество утилит для анализа последовательностей, формируя таким образом целостный программный пакет. EMBOSS - это хороший бесплатный вариант для начала работы с данными, содержащими последовательности. Для более глубокого анализа существуют коммерческие продукты с соответствующей поддержкой, техническим сервисом и обновлениями.

а) Скачайте и установите на компьютер последнюю версию EMBOSS (<ftp://emboss.open-bio.org/>). Документация на пакет доступна на <http://emboss.sourceforge.net/apps/#list> .

б) Найдите файл GenBank с последовательностью ДНК E.Coli isocitrate dehydrogenase J02799, который использовался в лабораторной работе №1.

в) Откройте файл и найдите в нём нуклеотидную и аминокислотную последовательности. Удостоверьтесь, что последовательность нуклеотидов -

ДНК - на самом деле кодирует аминокислотную последовательность белка.

Вы можете это сделать путем идентификации нуклеотидов (триплетов) в последовательности ДНК, которые точно соответствуют последовательности аминокислот. Будьте внимательны! Рамка считывания, (первый кодон) может начинаться не с первого нуклеотида, а со второго или с третьего!

- Генетический код может быть получен тут:

<http://helixweb.nih.gov/gcode.html>. Аминокислотный код представлен на

<http://www.mun.ca/biochem/courses/3107/aasymbols.html> Вам

может потребоваться убрать строку с комментариями, пробелы и иные символы из нуклеотидной последовательности.

- Быстрый, но в то же время грубый способ найти начало рамки считывания (то есть место в последовательности нуклеотидов, соответствующее первой аминокислоте) - это попытаться использовать какие-либо функции поиска подстроки. Мы знаем (см. генетический код), что ATG -М (метионин) и в то же время ATG может являться стартовым кодоном. Также можно попытаться найти и более специфичные (более длинные) подстроки, а не только последовательность, кодирующую М - попробуйте поискать кодоны для как минимум первых шести аминокислот в последовательности нуклеотидов. Можно также осуществить поиск конца последовательности -места, где заканчиваются кодоны, кодирующие аминокислоты белка. Нужно иметь в виду, что существуют три стоп-кодона - TAG, TAA, TGA. Эти кодоны не транслируются в аминокислоты. Вы можете использовать регулярные выражения (если это возможно), поскольку генетический код вырожден, то есть одна аминокислота может кодироваться несколькими кодонами. Например, регулярное выражение для поиска Lysine - AA[A,G] (для UNIX). Упражнение: как записать Leucine регулярным выражением?

г) Вторая проблема кроется в том, что последовательность, кодирующая интересующий нас белок, на самом деле может находиться на комплементарной нити ДНК (а не той, которая приведена в базе данных, так

называемые leading strand и lagging strand, strand + и strand-). Как можно получить комплементарную последовательность известного участка ДНК? Как известно, вторая нить, последовательность ДНК обладает свойством обратной комплементарности к первой:

5' C-A-T-G-T-C-C-A 3'

3' G-T-A-C-A-G-G-T 5'

Комплементарность в молекулярной биологии - это способность нуклеотидов образовывать только строго определенные пары (см. лекции).

Используйте программу revseq из пакета EMBOSS для:

- построения обратной последовательности;
- построения комплементарной последовательности;
- построения обратно-комплементарной последовательности.

Проделайте эти шаги для последовательности J02799.

д) Сколько возможных трансляций (в последовательность аминокислот) можно получить из одной последовательности нуклеотидов? Мы не будем пытаться получить все эти варианты.

Используйте программу transeq из EMBOSS для того, чтобы транслировать J02799 в возможную последовательность аминокислот. Эта программа может проделать это для любой из трех прямых и трех обратно-комплементарных последовательностей или для всех прямых и\или обратно-комплементарных последовательностей - для этого имеется флаг -frame.

Рамка считывания - это один из трех возможных путей считывания последовательности нуклеотидов в ДНК или РНК как серии неперекрывающихся троек нуклеотидов (триплетов) в зависимости от того, на каком нуклеотиде (первом, втором или третьем) началось чтение последовательности. Например, в TGCTGCTGC имеются следующие три возможные рамки считывания: TGC TGC TGC, GCT GCT и CTG CTG.

е) Проблема, описанная выше, заключается в выборе правильной рамки считывания для трансляции. К счастью, в случае J02799 имеется только один

из возможных шести вариантов, который даёт реальную и имеющую смысл ORF (open reading frame).

Используйте программу plotorf из EMBOSS для того, чтобы построить график всех возможных ORF в последовательности J02799 и идентифицировать самый длинный из них. Затем используйте эту информацию для определения начального и конечного нуклеотида в последовательности, которые имеют отношение к трансляции этого гена в соответствующий белок. Что вы можете сказать по поводу выбора ORF -правила, признаки etc? Важно помнить, что plotorf ищет участки, ограниченные старт- и стоп-кодонами. Иными словами, эукариотические экзоны, не имеющие старт-кодона, будут им пропущены. Plotorf имеет смысл применять на последовательностях ДНК прокариотов или на м-РНК эукариотов.

Справочная информация находится в

<http://emboss.bioinformatics.nl/cgi-bin/emboss/help/plotorf>

ж) Теперь давайте посмотрим на получаемые последовательности, которые похожи на J02799. Почему нам может быть это нужно и интересно?

- Эволюция последовательностей - из-за возникающих мутаций и генных перестроек\событий в разных организмах могут оказаться похожие последовательности;

- Поиск консервативных участков;

- Поиск записей с искусственно выполненными мутациями.

Для получения последовательностей, похожих на интересующий нас участок ДНК, мы будем использовать пакет программ BLAST - <http://www.ebi.ac.uk/blastall/> или

<http://www.ncbi.nlm.nih.gov/BLAST/>. Вы должны будете сохранять результаты в виде html-страниц - они потребуются нам в дальнейшей работе.

Выберите подходящую базу данных и программу (см. справку <http://www.ebi.ac.uk/Tools/sss/ncbiblast/help/index-protein.html>).

Используйте blastn в базе данных EMBL для поиска и сравнения нуклеотидных последовательностей и blastp в SwissProt для аминокислотных

последовательностей (белков).

Попробуйте найти последовательности, похожие на J02799.

а) Вирус H5N1 avian (птичий грипп) является очень вирулентным и опасным не только для птиц, но и для человека. Одним из его свойств является высокая мутационная активность.

Получите любую из последовательностей H5N1 в GenBank и затем при помощи BLAST найдите похожие последовательности. Каковы характеристики и локализация найденных последовательностей?

б) Даны три последовательности неизвестного происхождения и неизвестных функций:

- МРHK:

acauuugcuu cugacacaac uguguucacu agcaaccuca aacagacacc augguvcacc
ugacuccuga ggagaagucu gcgguuacug ccuguggggg caaggugaac guggaugaag uuggugguga
ggcccugggc aggcugcugg uggucuaccc uuggaccag agguucuug aguuccuugg ggauvcugcc
acuccugaug caguuauggg caaccuaagg ugaaggcuc auggcaaga agucvcggu gccuuuagug
auggccuggc ucaccuggac aaccucaagg gcaccuuvc cacacugagu gagvcgcacu gugacaagcu
gcacguggau ccugagaacu ucaggcuccu gggcaacgug cuggucugug uvcuggccca ucacuuuggc
aaagaauuca cccaccagu gcaggcugcc uaucagaaag ugguggcugg uguggcuau gcccuvgccc
acaaguauca cuaagvcgc uuucuvvcug uccaauuucu auuaaagguu ccuuuguucc

cuaaguccaa cuacuaaacu ggggggauauu augaagggcc uugagcaucu ggauucugcc
uaauaaaaaaaa cauuuuuuuuu cauugc

- сДНК:

acatttgctt ctgacacaac tgtgttctact agcaacctca aacagacacc atgggtgcacc tgactcctga
ggagaagtct gcggttactg ccctgtgggg caaggtgaac gtggatgaag ttggtgggtga ggccctgggc
aggctgctgg tggcttacct ttggaccag aggttctttg agtcctttgg ggatctgtcc actcctgatg cagttatggg
caaccctaag gtgaaggctc atggcaagaa agtgctcggt gcctttagt atggcctggc tcacctggac
aacctaagg gcacctttgc cactactgagt gagctgcact gtgacaagct gcacgtggat cctgagaact
tcaggctcct gggcaacgtg ctgggtctgtg tgctggccca tcactttggc aaagaattca ccccaccagt
gcaggctgcc tatcagaaag tgggtggctgg tgtggctaata gccctggccc acaagtatca

ctaagctcgc ttcttgctg tccaatttct attaaagggt cctttgttcc ctaagtccaa ctactaaact gggggatatt
atgaagggcc ttgagcatct ggattctgcc taataaaaaa cattatttt cattgc

- Трансляция в аминокислотную последовательность:

MVHLTPEEKSAVTALWGKVNDEVGGEALGRLLVVYPWTQRFFES
F GDLSTPDVAMGNPKVKANGKKVLGAFSDGLAHLDNLKGTFATLSEL
HCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGV
ANALANKYN

Используйте каждую из этих последовательностей для поиска при помощи BLAST. Оцените результаты поиска. В чём смысл полей Score, Evaluate и Identities (для ответа на вопрос нужно прокрутить страницу с результатами вниз, до строк, начинающихся с >)?

- Каким организмам принадлежат эти последовательности?
- Какова функция белка, закодированного в этом гене?
- Какие последовательности довольно близки к исследуемому участку ДНК? Каким организмам они принадлежат и какова их функция?
- Одинаковые ли результаты получились в трех поисковых запросах?
- Аккуратно удалите 21 нуклеотид из последовательностей мРНК и сДНК и соответствующие аминокислоты из последовательности белка. Изменило ли это результаты поиска BLAST?
- Аккуратно удалите некоторые нуклеотиды\аминокислоты в конце последовательностей и попробуйте провести BLAST снова.

Контрольные вопросы

1. Получили ли Вы те же результаты? Каковы значения scores и identities?
2. Сколько нуклеотидов\аминокислот нужно удалить для того, чтобы поиск перестал выдавать прежние результаты?
3. Что произойдёт, если Вы удалите нуклеотиды\аминокислоты из середины последовательностей?
4. Что произойдёт с трансляцией ДНК в аминокислотную последовательность, если Вы удалите 1\2\3 нуклеотида из середины

последовательности

ЛАБОРАТОРНАЯ РАБОТА № 3 «ПАРНОЕ ВЫРАВНИВАНИЕ»

Цель работы: изучение алгоритмов парного выравнивания.

Лабораторная работа № 2 была посвящена поиску по базам данных при помощи сервиса BLAST, который производит парное сравнение последовательностей и возвращает оценку выравнивания, E-value и собственно выровненные последовательности. BLAST может производить парное очень быстро. Например, поиск похожих последовательностей с использованием участка в 1568 нуклеотидов или 416 аминокислот из J02799 по базам последовательностей GenBank+EMBL+DDBJ+PDB (без EST, STS, GSS), то есть среди более чем 3,7 миллионов последовательностей (~16,5 млрд. нуклеотидов или аминокислот) производится за вполне приемлемое время. BLAST выполняет это, разумеется, не производя глобальное выравнивание, но при помощи определенных эвристических алгоритмов. Однако нужно понимать, что иные подходы могут быть иногда более эффективными, например, точные (не эвристические) методы, подходы, базирующиеся на глобальном выравнивании, глобальное и локальное выравнивание и даже точечные диаграммы - эффективные способы визуального сравнения двух последовательностей.

Выполнение работы

Мы будем производить сравнение последовательностей при помощи пакета EMBOSS:

- а) глобальные выравнивания (см. лекции); б) локальные выравнивания;
- в) работа с точечным графиком - визуализация выравнивания.

Для выполнения лабораторной работы вам понадобится пакет EMBOSS, установленный на локальном компьютере. Документация пакета доступна в сети Интернет.

Данные, необходимые для работы: последовательность белка

hemoglobin <http://en.wikipedia.org/wiki/Hemoglobin> (его v-chain, доступна по адресу <http://srs6.ebi.ac.uk/srs6bm/cgi-bm/wgetz?-e+|~swissprot-id:HBB HUMAN>)).

1. needle - программа, выполняющая глобальное выравнивание [http://en.wikipedia.org/wiki/Sequence alignment#Global alignment](http://en.wikipedia.org/wiki/Sequence_alignment#Global_alignment) по алгоритму Needleman-Wunsch [http://en.wikipedia.org/wiki/Needleman-Wunsch algorithm](http://en.wikipedia.org/wiki/Needleman-Wunsch_algorithm).

а) Вы можете использовать needle с параметрами для выполнения выравнивания и оценки его результатов, а также для генерации таблицы с оценками выравнивания, идентичности и похожести последовательностей и количеством вставленных промежутков для:

- человеческих последовательностей (по умолчанию);
- горилла;
- кролик;
- свинья.

б) Попробуем сконструировать филогенетическое дерево для человека, гориллы, кролика и свиньи на основе оценок глобального выравнивания (наподобие приведенного на рисунке 1). Для построения дерева нам понадобится создать таблицу 4x4 (таблица 1) и заполнить её оценками выравнивания (являющимися, в сущности, показателями эволюционного расстояния между последовательностями).

Gorilla H

unran Rabbit

Pig

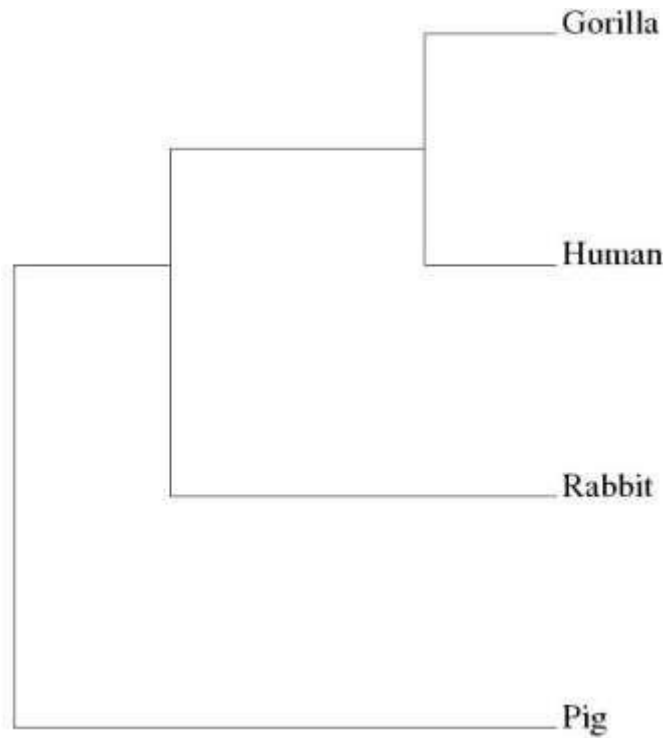


Рисунок 1. Пример укорененного филогенетического дерева Таблица 1.

	Human	Gorilla	Rabbit	Pig
Human	x	x	x	x
Gorilla		x	x	x
Rabbit			x	x
Pig				x

Вы также можете попробовать использовать значения sequence similarity (для белков), заполнив предлагаемую таблицу еще раз. Будет ли построено такое же дерево, как при использовании sequence identity?

2. Используя навыки, полученные в результате выполнения заданий лабораторной работы № 2, скачайте аминокислотные и нуклеотидные последовательности human isocitrate dehydrogenase. Используя needle,

проведите сравнение этих последовательностей и уже имеющейся isocitrate dehydrogenase от E.coli (J02799).

3. Повторите п.2, используя программу stretcher (алгоритм Myers-Miller,

работает быстрее, но менее точно, чем классический алгоритм). Отметьте различия в:

а) score, значениях sequence identity, sequence similarity, количестве вставленных промежутков, иные отличия;

б) времени выполнения (имеется ли достаточно сильная разница)?

4. Точечный график, по всей видимости, самый старый метод сравнения последовательностей (рисунок 2). Такой график - это визуальное представление похожих или идентичных участков в двух последовательностях. В таком представлении длина окна может оказаться фиксированной, также, как и длины двух последовательностей (но не всегда). Всякий раз, когда символ (или серия последовательно идущих символов -окно) одной последовательности идентичен\похож (в биологическом смысле) на символ из другой последовательности, на график наносится точка или короткая диагональ в соответствующей позиции.

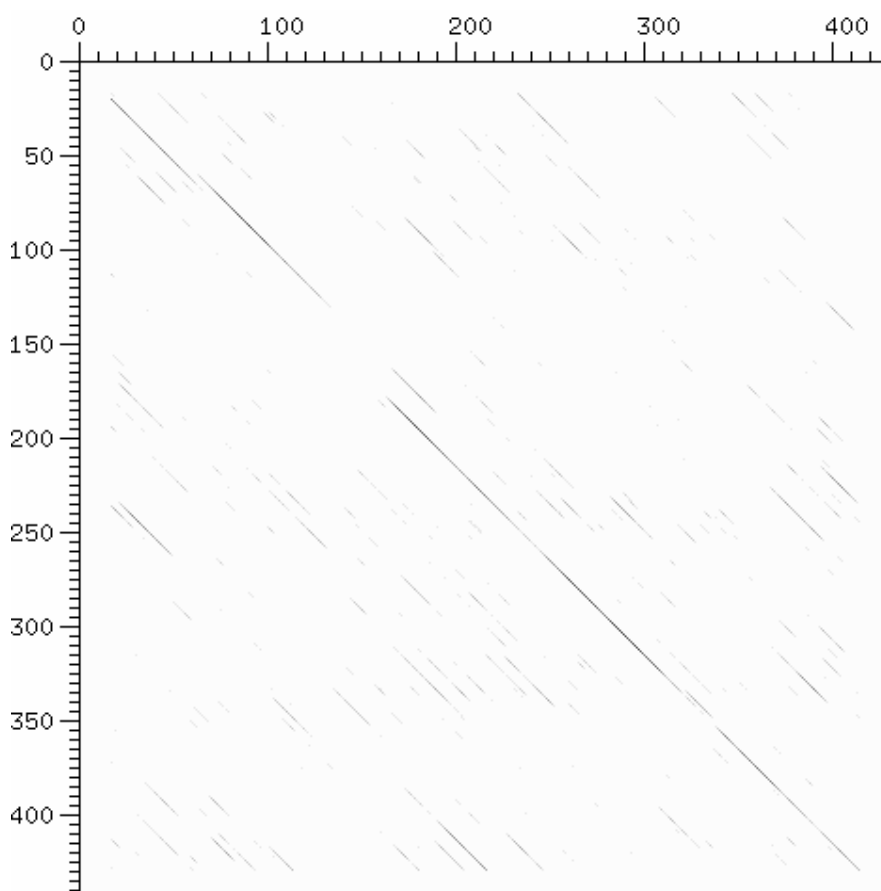


Рисунок 2. Точечный график для оценки идентичности двух последовательностей

Таким образом, когда две последовательности имеют похожие\идентичные участки по своей длине, диагональные линии, идущие от одного угла графика к противоположному, показывают эти участки. Это может быть главная диагональ (из района нуля и до конца обеих последовательностей), возможно, с перерывами, или отдельные линии в разных местах графика. Попробуйте объяснить физический и биологический смысл коротких линий, рассеянных по графику.

Программа dotmatcher отображает точечный график выравнивания двух последовательностей. Позволяет менять длину слова - минимальное количество подряд совпадающих символов.

Программа Dotter - более интерактивна по сравнению с предыдущей. Программа и документация: <http://bioinformatist.org/index.php/Dotter> и http://210.86.230.110/bioinfo/material/20070822_seq/Dotter%20binaries/Dotter%20info.pdf. Программы, строящие точечные графики, позволяют быстро найти консервативные и изменчивые участки в двух последовательностях.

Используйте одну из программ, строящих точечные графики, для визуального сравнения:

а) Isocitrate dehydrogenase человека и E.coli; б)

Haemoglobin от двух разных организмов; в) лобой последовательности с ней же;

г) последовательность с её обратной копией. Вам потребуется программа revseq (см. лабораторную работу № 2);

д) соедините в текстовом редакторе две очень разных последовательности в различном порядке (вопрос по ходу: можно ли соединить нуклеотидную и аминокислотную последовательности? Например, соедините последовательности haemoglobin (H) и isocitrate dehydrogenase (I) в порядке HI и IH. Сравните получившиеся синтетические последовательности (называемые в биологии химерами) с использованием метода глобального и локального выравнивания (в соответствующих программах);

е) попробуйте сделать это для:

- нуклеотидной последовательности;
- аминокислотной последовательности. ж) в чём разница выравниваний?

з) Попробуйте изменить параметр `threshold`. Получится ли получить лучшее представление совпадающих регионов.

5. Программа `water` и `matcher` рассчитывают локальное выравнивание (http://en.wikipedia.org/wiki/Sequence_alignment#Local_alignment) - поиск похожих\идентичных регионов (локальных участков) в двух (в случае парного выравнивания) последовательностях по всей их длине. Методы локального выравнивания бывают очень полезны для поиска по базам данных и для решения других задач, например, для поиска небольших идентичных или похожих участков у белков - доменов.

Используйте эти программы для выравнивания некоторых из последовательностей, с которыми сегодня работали. Получилось ли обнаружить качественное локальное выравнивание?

Контрольные вопросы

1. Основные алгоритмы построения филогенетических деревьев - их достоинства и недостатки. UPGMA и NJ (их отличия), максимальной бережливости (*maximal parsimony*), максимального правдоподобия, минимальной эволюции.
2. Группы аминокислот. Группировка аминокислот с эволюционной и структурной точек зрения.
3. Открытая рамка считывания. Её нахождение и транскрипция. Поиск открытых рамок считывания - методы.
4. Биоинформатика и филогенез. Молекулярные часы. Клада, OTU, ветвь, лист, корень. Ультраметрическое и неультраметрическое дерево. Ортологи, паралогии, гомологи, ксенологи.
5. Определение филы, таксона, вида. Филогенетические деревья видов и генов, их различия.
6. Какой тест может быть проведен на филогенетическом дереве,

построенном по NJ алгоритму, чтобы высказать предположение о его корректности.

Приложения к лабораторной работе № 3

Globin Sequences (Beta-Chain)

Human HEMOGLOBIN BETA CHAIN.(HBB_HUMAN) SEQUENCE 146 AA; 15867 MW; VHLTPEEKSAVTALWGKVVDEVGGEALGRLLVVYPWTQRFFESF

STPDAVMGNPKVKAHGKKVLGAFSDGLAHLNKGTFATLSELHC

GDL

DKL

HVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANAL

AH KYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB HUMAN\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB HUMAN]) Human sickle beta-hemoglobin SEQUENCE 147 AA MVHLTPEEKSAVTAXWGKVVDEVGGEALGRLLVVYPWTQRFFES

FG DLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLNKGTFATLSELHCD

KLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANAL

AHKYH

<http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=protein&dopt=>

Ge

[nPept&list uids=183945](#)

Gorilla gorilla gorilla (Lowland gorilla).HEMOGLOBIN BETA CHAIN.(HBB_GORGO) SEQUENCE 146 AA; 15839 MW; VHLTPEEKSAVTALWGKVVDEVGGEALGRLLVVYPWTQRFFESF

STPDAVMGNPKVKAHGKKVLGAFSDGLAHLNKGTFATLSELHC

GDL

DKL

HVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANAL

AH KYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB GORGO\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB GORGO])

Hylobates lar (Common gibbon).HEMOGLOBIN BETA

CHAIN.(HBB_HYLLA) SEQUENCE 146 AA; 15925 MW;

VHLTPEEKSAVTALWGKVVNDEVGGEALGRLLVVYPWTQRFFESF

STPDVAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFQAQLSELHC

GD L DK

LHVDPENFRLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANA

LA HKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB HYLLA\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB HYLLA]) Pres-

bytis entellus (Hanuman langur).HEMOGLOBIN BETA CHAIN.(HBB_PREEN)

SEQUENCE 146 AA; 15895 MW; VHLTPEEKAAVTALWGKVVNDEVGGE-

ALGRLLVVYPWTQRFFESF

GD

LSSPDVAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFQAQLSELH

CD

KLHVDPENFRLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVAN

AL AHKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB SEMEN\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB SEMEN]) Colo-

bus badius (Red colobus).HEMOGLOBIN BETA CHAIN.(HBB_COLBA) SE-

QUENCE 146 AA; 15870 MW; VHLTPDEKNAVTALWGKVVNDEVGGE-

ALGRLLVVYPWTQRFFDSF

GD

LSTADVAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFQAQLSELH

CD

KLHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVAN

AL AHKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB COLBA\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB COLBA])

Oryctolagus cuniculus (Rabbit).HEMOGLOBIN BETA-1 AND BETA-2

CHAINS.(HBB_RABIT) SEQUENCE 146 AA; 16001 MW;

VHLSSEEKSAVTALWGKVNVEEVGGEALGRLLVVYPWTQRFFESFG
DL DKL SSANAVMNNPKVKAHGKKVLA AFSEGLSHLDNLKGTFAKLSELHC
HVDPENFRL LGNVLVIVLSHHFGKEFTPQVQAAYQKVVAGVANAL
AHK YH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB_RABIT\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB_RABIT]) Bison
bonasus (European bison).HEMOGLOBIN BETA CHAIN.(HBB_BISBO) SE-
QUENCE 145 AA; 15976 MW; MLTAEKAAVTAFWGKVHVDEVGGEAL-
GRLLVVYPWTQRFFESFG

DL
SSADAVMNNAKVKAHGKKVLDSFSNGMKHLDDLKGTFAALSELHC

DK
LHVDPENFKLLGNVLVVVLARHFGKEFTPVLQADFQKVVTGVANA

LAH RYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB_BISBO\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB_BISBO]) Sus
scrofa (Pig).HEMOGLOBIN BETA CHAIN.(HBB_PIG) SEQUENCE

146 AA; 16034 MW; VHLSAEEKEAVLGLWGKVNDEVGGEAL-
GRLLVVYPWTQRFFESF

GD
LSNADAVMGNPKVKAHGKKVLQSFSDGLKHLDNLKGTF AKLSELH

CD
QLHVDPENFRL LGNVIVVVLARRLGHDFNP NVQAAFQKVVAGVAN

AL AHKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB_PIG\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB_PIG]) Lutra lu-
tra (European river otter).HEMOGLOBIN BETA CHAIN.(HBB_LUTLU) SE-
QUENCE 146 AA; 15950 MW; VHLTGEKAAVTSLWGKVNDEVGGE-
ALGRLLVVYPWTQRFFDSF

GD
LSSPDAVMGNPKVKAHGKKVLNSFSEGLKNLDNLKGTF AKLSELHC

DK

LHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANA

LA HKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB LUTLU\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB LUTLU]) Thero-
pithecus gelada (Gelada baboon).HEMOGLOBIN BETA CHAIN.(HBB_THEGE)
SEQUENCE 146 AA; 15925 MW; VHLTPEEKNAVTTTLWGKVNVDDEVGGE-
ALGRLLVVYPWTQRFFDSF

GD

LSSPAAVMGNPKVKAHGKKVLGAFSDGLNHLDNLKGTFAQLSELH

CD

KLHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVAN

AL AHKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB THEGE\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB THEGE])

ЛАБОРАТОРНАЯ РАБОТА № 4 «МНОЖЕСТВЕННОЕ ВЫРАВНИВАНИЕ»

Цель работы: выполнение множественные выравнивания последовательностей.

Выполнение работы

Программа, в которой обычно выполняется выравнивание - ClustalW. Документацию по этому инструменту можно найти на <http://www.ebi.ac.uk/2can/tutorials/nucleotide/clustalw.html>.

Прошлые лабораторные работы познакомили нас с методиками сравнения последовательностей, базирующимися на локальном и глобальном выравнивании. На этот раз мы будем использовать различные инструменты и различные матрицы оценок для этого.

Для чего нужно Multiple sequence alignment (MSA)? http://en.wikipedia.org/wiki/Multiple_Sequence_Alignment

Множественное выравнивание может быть использовано при изучении эволюционных отношений между белками или последовательностями ДНК. Так как изменения в нуклеотидных последовательностях в процессе эволюции накапливаются, то можно проанализировать гомологичные гены (гены, имеющие одно эволюционное начало) от разных организмов и затем сравнить их при помощи MSA. Результаты такого выравнивания могут быть для определения консервативных, сохранившихся, общих областей, то есть областей, изменение которых может повлиять на функцию. Это бывает очень полезно для планирования экспериментов определения и модификации функции определенных белков, для предсказания структуры и функции белка и для определения новых членов известных белковых семейств (то есть для классификации).

MSA бывает полезна для представления свойств набора (семейства) последовательностей. Эти свойства (дескрипторы) могут в дальнейшем использоваться в качестве шаблонов для поиска по базам данных

последовательностей, похожих на имеющийся набор (bootstrapping).

MSA программы и техники Прогрессивные стратегии для MSA:

Общий подход (эвристический) для множественного выравнивания - это последовательное выравнивание пар последовательностей. Все техники этого подхода подразумевают выполнение двух этапов: попарное выравнивание всех последовательностей с последующей кластеризацией, с построением guide tree и выравнивание результатов парных выравниваний в соответствии с guide tree.

Одна из самых популярных программ для MSA - ClustalW (и ее модификации). Она использует вышеописанный подход для выполнения MSA последовательностей ДНК или белков. Она определяет лучшее парное выравнивание и помещает его на первые позиции, выделяет цветом и т.д. Программа также генерирует филограмму (филогенетическое дерево) и кладограмму (граф сестринского соответствия между известными группами без учёта фактора времени в эволюции). Это может быть полезно для изучения эволюционных взаимосвязей в наборе последовательностей.

1. Для выполнения MSA используйте ClustalW на сайте EBI
<http://www.ebi.ac.uk/Tools/msa/clustalw2/>. Самый простой формат

последовательностей при использовании ClustalW - FASTA

<http://www.ebi.ac.uk/Tools/sss/fasta/>:

>SEQUENCE_NAME1 PLUS ANY OTHER COMMENTS

SEQUENCE1 (может быть в несколько строк) >SEQUENCE_NAME2 PLUS
ANY OTHER COMMENTS SEQUENCE2 (может быть в несколько строк)

>SEQUENCE_NAME_n PLUS ANY OTHER COMMENTS

SEQUENCE_n (может быть в несколько строк)

Можно также использовать ClustalW локально в виде командной строки (на UNIX) или многочисленные реализации для Windows.

а) Запустите ClustalW (в случае локального использования - без

параметров)

б) Выберите тип последовательности Protein.

в) Загрузите последовательности в форму в формате Fasta передайте на обработку fasta-файл с диска (поле Upload).

г) Установите переключатель Slow/Fast в Slow - полный точный MSA. д) Вы можете получить результаты по e-mail, отметив соответствующий checkbox или увидеть их на открывшейся веб-странице.

Используйте последовательности белков из семейства глобинов, представленные в конце лабораторной работы (как список или fasta-файл). Символы консенсуса:

Полученное выравнивание будет содержать следующие символы под каждым блоком последовательностей, показывающие степень консервации символа в каждой колонке:

* - все аминокислоты/нуклеотиды в этой колонке идентичны во всех последовательностях;

: - имеются консервативные замены (см. таблицу цветов на странице);

. - имеются полуконсервативные замены. Вам

нужно обратить внимание на:

а) Файл выравнивания (лучше с цветами);

б) Филограмму и кладограмму (отметьте разницу между ними);

в) Выравнивание в редакторе Jalview (может быть запущен на странице результатов ClustalW <http://www.jalview.org/>).

Вы также можете построить выравнивание других интересующих вас последовательностей, найденных в базе NCBI (<http://www.ncbi.nlm.nih.gov/>) и сохраненных в fasta-формате. Например:

- если вас интересуют последовательности вируса птичьего гриппа, выполните поиск по H5N1. В случае ВИЧ - поиск по HIV-2;

- вам потребуется выбрать последовательности (базы данных), с которыми вы хотите поработать - это либо нуклеотидные либо аминокислотные последовательности и соответствующие базы данных;

- установите режим отображения FASTA;
- сохраните данные в текстовом файле;
- отредактируйте сохраненный файл: удалите все пробелы из первой строки с комментариями и иные символы, которые вы не хотите видеть в результатах выравнивания и убедитесь в том, что строки комментариев для каждой последовательности в файле уникальна, при необходимости отредактируйте эти строки для того, чтобы последовательности можно было различать.

В конце лабораторной работы - последовательности H5N1 с их локализацией, видовой принадлежностью, местом обнаружения вируса и прочими данными.

Вы также можете использовать одну интересующую вас последовательность (например, глобин) и затем использовать BLAST для поиска похожих последовательностей. После этого вы можете сохранить все или некоторые из них (например, от различных видов) в формате fasta и использовать полученный файл для MSA.

Самостоятельно изучите 2can ClustalW tutorial (<http://www.ebi.ac.uk/2can/tutorials/nucleotide/clustalw.html>).

Сейчас мы рассмотрим выравнивание нескольких нуклеотидных последовательностей гена белка тропомиозина (<http://www.biology-online.org/dictionary/Tropomyosin>), номера доступа которого приведены ниже:
BF056441 BE8487196 BF022813 BF452255 BG089808 BG147728 BI817778
AF186109 AF186110 AF310722 AF362886 AF362887 AF087679
SSAJ803 SSAJ804

Эти последовательности (в конце лабораторной работы) имеют больше различий, чем исследованные нами глобины.

Попробуйте использовать другие программы для MSA, выполнив выравнивания на ваших данных.

- T-coffee (<http://www.igs.cnrsmr.fr/Tcoffee/tcoffee.cgi/index.cgi>) - эта программа выполняет более точное выравнивание по сравнению с ClustalW

для последовательностей с идентичностью < 30%. Работает медленнее;

- Muscle (http://phylogenomics.berkeley.edu/cgi-bin/muscle/input_muscle.py) - результаты по e-mail. Если вы хотите использовать пакет EMBOSS:

Документация по EMBOSS доступна на <http://emboss.sourceforge.net/>; Пакет EMBOSS доступен как online, так и для использования локально.

2. Исследование из лабораторной работы № 3.

Программы water и matcher из пакета EMBOSS выполняют локальное выравнивание двух последовательностей и находят в них похожие участки. ([http://en.wikipedia.org/wiki/Sequence alignment#Local alignment](http://en.wikipedia.org/wiki/Sequence_alignment#Local_alignment)) Выравниваемые последовательности могут быть разной длины. Методы локального выравнивания могут быть очень полезны при поиске в базах данных и других случаях, например, при поиске наиболее совпадающих небольших участков совпадений, например, доменов в белках.

Используя эти программы, проведите выравнивание последовательностей, приведенных ниже. Удалось ли выявить хорошее локальное выравнивание?

3. Программы, полезные для поиска последовательностей (таблица 2), можно найти на EBI tools page (<http://www.ebi.ac.uk/Tools/index.html>). Попробуйте поработать с некоторыми из них. Сравните результаты их работы. Одинаковы ли они при одном и том же запросе\последовательности? Оцените время работы различных приложений для поиска последовательностей и результаты\подробности в отчетах. Подробная online помощь по вопросам гомологии и сравнению последовательностей - <http://www.ebi.ac.uk/Tools/sss/> .

Таблица 2.

Fasta3 http://www.ebi.ac.uk/Tools/ss/s/fasta/	Sequence similarity and homology searching against nucleotide and protein database using Fasta3
WU-Blast2 http://www.ebi.ac.uk/Tools/ss/s/wublast/	Washington University blast2 (blast 2.0 with gaps)
NCBI-Blast2 http://www.ebi.ac.uk/Tools/ss/s/ncbiblast/	NCBI blast2 (blastall) program
MPsrch http://www.ebi.ac.uk/Tools/M_Psrch/	Edinburgh University's new implementation of the Smith and Waterman algorithm
Scanps2.3 http://www.ebi.ac.uk/Tools/scanps/	Version 2.3 of Scanps. Fast implementation of the true Smith & Waterman algorithm for protein database searches

4. Попробуйте провести различные поиски и выравнивания последовательностей по базам данных, используя при этом различные матрицы расстояний. Online помощь по выбору матриц -<http://www.ebi.ac.uk/2can/tutorials/matrices.html> .

5. Повторите поиск, используя BLAST с различным expected thresholds. Это дает возможность уменьшить или увеличить список совпадающих последовательностей\участков. Online помощь по использованию BLAST - <http://blast.ncbi.nlm.nih.gov/Blast.cgi> .

6. Вы можете производить поиск отдельно по целым геномам или протеомам, используя сервер Proteomes & Genomes Fasta3 <http://www.ebi.ac.uk/Tools/sss/fasta/>.

7. Попробуйте поэкспериментировать с ним.

Приложение к лабораторной работе № 4

Последовательности и материалы, необходимые для выполнения лабораторной работы.

Globin Sequences (Beta-Chain)

Human HEMOGLOBIN BETA CHAIN.(HBB_HUMAN) SEQUENCE 146 AA;

15867 MW;

VHLTPEEKSAVTALWGKVNDEVGGEALGRLLVVYPWTQRFFESFGDL

STPDAMGPNPKVKAHGGKVLGA FSDGLAHL DNLKGTFATLSELHCDKL

HVDPENFRL LGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANALAH KYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB HUMAN\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB HUMAN]) Human

sickle beta-hemoglobin SEQUENCE 147 AA

MVHLTPVEKSAVTAXWGKVNDEVGGEALGRLLVVYPWTQRFFESFG

DLSTPDAMGPNPKVKAHGGKVLGA FSDGLAHL DNLKGTFATLSELHCD

KLHVDPENFRL LGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANAL AHKYH

<http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=protein&dopt=GenPept&listuids=183945>

Gorilla gorilla gorilla (Lowland gorilla).HEMOGLOBIN BETA CHAIN. (HBB_GORGO)

SEQUENCE 146 AA; 15839 MW; VHLTPEEKSAVTALWGKVNDEVGGEAL-

GRLLVVYPWTQRFFESFGDL STPDAMGPNPKVKAHGGKVLGA FSDGLAHL D-

NLKGTFATLSELHCDKL HVDPENF-

KLLGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVANALAH KYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB GORGO\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB GORGO]) Hylobates

lar (Common gibbon).HEMOGLOBIN BETA CHAIN.(HBB_HYLLA) SEQUENCE

146 AA; 15925 MW; VHLTPEEKSAVTALWGKVNDEVGGEAL-

GRLLVVYPWTLQRFFESFGDL

STPDAVMGNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDK
LHVDPENFRLLGNVLVCVLAHHFGKEFTTPQVQAAYQKVVAGVANALA HKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[swissprot-id:HBB_HYLLA\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[swissprot-id:HBB_HYLLA]) Presbytis
entellus (Hanuman langur).HEMOGLOBIN BETA CHAIN.(HBB_PREEN) SE-
QUENCE 146 AA; 15895 MW; VHLTPEEKAAVTALWGKVNVDDEVGGEAL-
GRLLVVYPWTQRFFESFGD LSSPDAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFAQLSELHCD

KLHVDPENFRLLGNVLVCVLAHHFGKEFTTPQVQAAYQKVVAGVANAL AHKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB_SEMEN\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB_SEMEN]) Colobus
badius (Red colobus).HEMOGLOBIN BETA CHAIN.(HBB_COLBA) SEQUENCE
146 AA; 15870 MW; VHLTPDEKNAVTALWGKVNVDDEVGGEAL-
GRLLVVYPWTQRFFDSFGD LSTADAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFAQLSELHCD

KLHVDPENFRLLGNVLVCVLAHHFGKEFTTPQVQAAYQKVVAGVANAL AHKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB_COLBA\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB_COLBA]) Oryctolagus
cuniculus (Rabbit).HEMOGLOBIN BETA-1 AND BETA-2 CHAINS.(HBB_RABIT)
SEQUENCE 146 AA; 16001 MW; VHLSSSEKSAVTALWGKVNVEEVGGEAL-
GRLLVVYPWTQRFFESFGDL SSANAVMNNPKVKAHGKKVLAAFSEGLSHLD-
NLKGTFAKLSELHCDKL HVDPENFRLLGNVLVI-
VLSSHFGKEFTTPQVQAAYQKVVAGVANALAHK YH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB_RABIT\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB_RABIT]) Bison
bonasus (European bison).HEMOGLOBIN BETA CHAIN.(HBB_BISBO) SE-
QUENCE 145 AA; 15976 MW; MLTAEKAAVTAFWGKVHVDEVGGEAL-
GRLLVVYPWTQRFFESFGDL SSADAVMN-
NAKVKAHGKKVLDSFSNGMKHLDDLKGTFAALSELHCDK

LHVDPENFRLLGNVLVVVLARHFGKEFTPVLQADFQKVVVTGVANALAH RYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB_BISBO\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB_BISBO])

Sus scrofa (Pig).HEMOGLOBIN BETA CHAIN.(HBB_PIG) SEQUENCE 146 AA; 16034 MW;
VHLSAEEKEAVLGLWGKVNVDDEVGGEALGRLLVVYPWTQRFFESFGD LSNADAVMGNPKV-
KAHGKKVLQSFSDGLKHLNLDNLKGTFAKLSELHCD QLHVDPENFRLLGNVIVVVLARRLGH-
DFNPNVQAAFQKVVAGVANAL AHKYH

[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB_PIG\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB_PIG]) Lutra

lutra (European river otter).HEMOGLOBIN BETA CHAIN.(HBB_LUTLU)

SEQUENCE 146 AA; 15950 MW;

VHLTGEEKAAVTSLWGKVNVDDEVGGEALGRLLVVYPWTQRFFDSFGD

LSSPDAVMGNPKVKAHGKKVLNSFSEGLKNLDNLKGTFAKLSELHCDK
LHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANALA HKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB LUTLU\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB LUTLU]) Thero-
pithecus gelada (Gelada baboon).HEMOGLOBIN BETA CHAIN.(HBB_THEGE)
SEQUENCE 146 AA; 15925 MW; VHLTPEEKNAVTTTLWGKVNVDDEVGGEAL-
GRLLVVYPWTQRFFDSFGD LSSPAAVMGNPKVKAHGKKVL-
GAFSDGLNHLNLDNLKGTFQAQLSELHCD
KLHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANAL AHKYH
[http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+\[~swissprot-id:HBB THEGE\]](http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-e+[~swissprot-id:HBB THEGE])

Глобины в формате FASTA:

>Human VHLTPEEKSAVTALWGKVVNDEVGGEAL-
GRLLVVYPWTQRFFESFGDL STPDAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFATLSELHCDKL
HVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAH KYH
>Human_sickle MVHLTPEEKSAVTAXWGKVVNDEVGGEAL-
GRLLVVYPWTQRFFESFG DLSTPDAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFATLSELHCD
KLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANAL AHK-
YH
>Gorilla VHLTPEEKSAVTALWGKVVNDEVGGEAL-
GRLLVVYPWTQRFFESFGDL STPDAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFATLSELHCDKL HVDPENF-
KLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAH KYH
>Common_gibbon VHLTPEEKSAVTALWGKVVNDEVGGEAL-
GRLLVVYPWTQRFFESFGDL STPDAVMGNPKVKAHGKKVL-
GAFSDGLAHLNLDNLKGTFQAQLSELHCDK
LHVDPENFRLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANALA
HKYH
>Hanuman_langur VHLTPEEKAAVTALWGKVVNDEVGGEAL-
GRLLVVYPWTQRFFESFGD LSSPDAVMGNPKVKAHGKKVLGAFSDGLAHL-
NLKGTFQAQLSELHCD
KLHVDPENFRLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANAL AHKYH
>Red_colobus

VHLPDEKNAVLTALWGKVNVEVGGEALGRLLVVYPWTQRFFDSFGD
LSTADAVMGNPKVKAHGKKVLGAFSDGLAHLNLDLKGTFQALSELHCD
KLHVDPENFKLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANAL AHKYH
>Rabbit VHLSSEEKSAVTALWGKVNVEVGGEALGRLLVVYPWTQRFFESFGDL
SSANAVMNNPKVKAHGKKVLAASFSEGLSHLDNLKGTFQAKLSELHCDKL
HVDPENFRLLGNVLVIVLSHHFGKEFTPQVQAAYQKVVAGVANALAHK YH
>European_bison MLTAEKAAVTAFWGKVHVDEVGGEAL-
GRLLVVYPWTQRFFESFGDL SSADAVMN-
NAKVKAHGKKVLDSFSNGMKHLDDLKGTFQALSELHCDK LHVDPENF-
KLLGNVLVVVLARHFGKEFTPVLQADFQKVVTVGVANALAH RYH
>Pig
VHLSAEEKEAVLGLWGKVN DEVGGEALGR LLVVYPWTQR FFESFGDLSN
ADAVMGNPKV KAHGKKVLQS FSDGLKHLNLDLKGTFQAKLSE-
LHCDQLHVDPENFRLLGNVIVVVLARRLG
HDFNPNVQAQFQKVVAGVANALAHKYH
>European_river_otter VHLTGEEKAAVTSLWGKVNVEVGGEAL-
GRLLVVYPWTQRFFDSFGD LSSPDVAVMGNPKVKAHGKKVLNSFSEGLK-
NLDNLKGTFQAKLSELHCDK LHVDPENF-
KLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANALA HKYH
>Gelada_baboon VHLPPEKNAVTTLWGKVNVEVGGEAL-
GRLLVVYPWTQRFFDSFGD LSSPAVAVMGNPKVKAHGKKVLGAFSDGLNHL-
NLKGTFQALSELHCD KLHVDPENF-
KLLGNVLVCVLAHHFGKEFTPQVQAAYQKVVAGVANAL AHKYH

H5N1 (вирус птичьего гриппа) в формате Fasta:

>1A_Cygnus_olor_Astrakhan_Ast05_2_3_2005
>2A_Goose_Guangdong_1_96
>3A_Hong_Kong_1073_99
>4A_chicken_Hebei_326_2005
>5A_chicken_Hebei_718_2001
>6A_chicken_Hebei_326_2005
>7A_chicken_Hebei_718_2001
>8A_chicken_Hebei_326_2005
>9A_chicken_Hebei_108_02

Последовательности тропомиозинов.

- >embl:BF056441 BF056441; 7k05a04.x1 NCI_CGAP_GC6 Homo sapiens
cDNA clone IMAGE:3443238 3' similar to SW:TPM4_HUMAN P07226 TROPOMYOSIN, FIBRO-
BLAST NON-MUSCLE TYPE; mRNA sequence.
- >embl:BE848719 BE848719; uw40c07.y1 Soares_thymus_2NbMT Mus musculus cDNA clone IM-
AGE:3419148 5' similar to SW:TPM4_HUMAN P07226 TROPOMYOSIN, FIBROBLAST
NON-MUSCLE TYPE ; mRNA sequence.
- >embl:BF022813 BF022813; uw40c07.x1 Soares_thymus_2NbMT Mus musculus cDNA clone IM-
AGE:3419148 3' similar to SW:TPM4_RAT P09495 TROPOMYOSIN 4, EMBRYONIC FIBRO-
BLAST ISOFORM ; mRNA sequence.
- >embl:BF452255 BF452255; uz86d11.y1 NCI_CGAP_Lu29 Mus musculus cDNA clone IM-
AGE:3675957 5' similar to SW:TPM4_RAT P09495 TROPOMYOSIN 4, EMBRYONIC FIBRO-
BLAST ISOFORM ; mRNA sequence.
- >embl:BG089808 BG089808; mab82b11.x1 NCI_CGAP_BC3 Mus musculus cDNA clone IM-
AGE:3976676 3' similar to SW:TPM4_HUMAN P07226 TROPOMYOSIN, FIBROBLAST
NON-MUSCLE TYPE ; mRNA sequence.
- >embl:BG147728 BG147728; mab53f06.x1 Soares_NMEBA_branchial_arch Mus musculus cDNA
clone IMAGE:3974147 3' similar to SW:TPM4_HUMAN P07226 TROPOMYOSIN, FIBROBLAST
NON-MUSCLE TYPE; mRNA sequence.
- >embl:BI817778 BI817778; G3-F20 Axolotl Lambda Zap Library
Ambystoma mexicanum cDNA similar to Homo sapiens gblAAG17014.1IAF186109_1 (2.0e-40),
TPM4-ALK fusion oncoprotein type 2, mRNA sequence.
- >embl:AF186109 AF186109; Homo sapiens TPM4-ALK fusion oncoprotein type 2 (TPM4-ALK fusion)
mRNA, partial cds.
- >embl:AF186110 AF186110; Homo sapiens TPM4-ALK fusion oncoprotein type 1 (TPM4-ALK fusion)
mRNA, partial cds.
- >embl:AF310722 AF310722; Homo sapiens tropomyosin 4-anaplastic lymphoma kinase fusion protein
(TPM4-ALK) mRNA, partial cds.
- >embl:AF362886 AF362886; Homo sapiens tropomyosin 4-anaplastic lymphoma kinase fusion protein
major isoform mRNA, partial cds.
- >embl:AF362887 AF362887; Homo sapiens tropomyosin 4-anaplastic lymphoma kinase fusion protein
minor isoform mRNA, partial cds.
- >embl:AF087679 AF087679; Sus scrofa tropomyosin 4 (TPM4) mRNA, complete cds.

Контрольные вопросы

1. Редакционное расстояние между двумя последовательностями. Сложность наивного алгоритма его определения.
2. Дано: последовательности WATER и WINE. Скоринг: match- 5, mismatch- -5, вставка промежутка (gap insertion)- -1. Построить таблицу выравнивания и найти по ней путь для него.
3. Локальное выравнивание, задачи, примеры.
4. Множественное выравнивание.
5. Третичная структура белка. Фолдинг.
6. Предсказание третичной структуры белка. Моделирование гомологов.

Методы, ПО и сервисы

ЛАБОРАТОРНАЯ РАБОТА № 5 «ИЗУЧЕНИЕ ШАБЛОНОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ»²

Цель: получение опыта исследований и поиска шаблонов (паттернов) последовательностей.

Выполнение работы

Базы данных и инструменты поиска:

а) PROSITE (<http://ca.expasy.org/prosite/>) - база данных доменов и семейств белков. Она содержит биологически важные сайты, мотивы и профили, которые помогают довольно четко определить семейство (если таковое имеется), к которому принадлежит новая последовательность белка.

б) Приложение ScanProsite (<http://ca.expasy.org/tools/scanprosite/>) позволяет пользователю сканировать последовательности белков (в базе данных UniProt-SwissProt/TrEMBL, PDB или пользовательские последовательности) на наличие мотивов, профилей или шаблонов (паттернов), имеющихся в базе данных PROSITE или производить поиск по белковым базам данных в поисках определенных мотивов.

Большинство программ, используемых в этой лабораторной работе, находится на <http://ca.expasy.org/tools/#pattern>.

Некоторые программы, которые могут использоваться для поиска и изучения шаблонов\мотивов (<http://bioweb.pasteur.fr/seqanal/motif/intro-uk.html>):

а) PRATT на EBI, <http://ca.expasy.org/tools/pratt/> или PrattWWW (<http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::pratt>): инструмент для поиска мотивов (I. Jonassen). Руководство пользователя на Pratt находится по адресу <http://www.ebi.ac.uk/Tools/pratt/>. Pratt ищет повторяющиеся мотивы, может генерировать шаблоны - регулярные выражения в синтаксисе ProSite и может находить шаблоны и паттерны, являющимися важными, определяющими для введенных последовательностей. Кроме того, программа может улучшать,

оптимизировать паттерны и шаблоны, которые находит.

б) MEME <http://meme.sdsc.edu/meme/meme-input.html> (Multiple EM for Motif Elicitation): поиск мотивов (высококонсервативных образцов) в группах (наборах) последовательностей гомологичных ДНК или белков (Т. Bailey, С. Elkan, В. Grundy).

в) PFTOOLS (<http://mobylye.pasteur.fr/cgi-bin/portal.py?#forms::pftools>): PRO-FILE tools (Р. Bucher).

г) SMILE (http://www-igm.univ-mlv.fr/~marsan/smile_english.html): Structured Motif Inference and Evaluation (L. Marsan, J. Allali).

Простые, структурные и распределенные мотивы
<http://mobylye.pasteur.fr/cgi-bin/portal.py?#forms::smile>;

а) Prophet (<http://mobylye.pasteur.fr/cgi-bin/portal.py?#forms::prophet>) веб-сервер: выравнивание профилей, возможно с промежутками. EMBOSS-prophet ([http://embos.s.bioinformatics.nl/cgi-](http://embos.s.bioinformatics.nl/cgi-bin/emboss/prophet)

[bin/emboss/prophet](http://embos.s.bioinformatics.nl/cgi-bin/emboss/prophet)) - как и весь EMBOSS, может быть запущен локально; б)

Consensus (<http://mobylye.pasteur.fr/cgi-bin/portal.py?#forms::consensus>): Определение консенсусных

шаблонов в невыровненных последовательностях ДНК и белков. Развитая статистическая база для штрафов за промежутки (G.Z.Hertz и G.D.Stormo).

в) PatternHunter - <http://www.bioinformaticssolutions.com/all-products/ph> Дополнительные ресурсы для поиска мотивов и шаблонов:
<http://bioweb2.pasteur.fr/motif/>

Цитохромы Р450 - семейство наиболее мощных детоксицирующих ферментов организма. Известно более 60-и ключевых форм с сотнями возможных генетических вариаций, выливающих в огромный спектр восприимчивости к различным токсинам. Дополнительную информацию по Р450 можно найти по ссылке <http://www.mall-net.com/mcs/p450.html>. Биохимики высказали предположение (благодаря интуиции и опыту), что последовательности белка Р450 может быть описана (характеризована)

следующим мотивом: FMFEGHDTTA.

Этот мотив был обнаружен в наборе последовательностей Цитохрома P450 (последовательности в конце лабораторной работы).

Однако, возник вопрос, нет ли других мотивов\паттернов, которые могут характеризовать и выделять последовательности цитохромов P450 из других семейств.

В этом реальном исследовании нужно:

а) Доказать, что мотив FMFEGHDTTA действительно является уникальным и может идентифицировать семейство P450 - это значит, что можно использовать указанный мотив для поиска по базе ProSite для идентификации;

б) Использовать любую из найденных последовательностей для того, чтобы попытаться извлечь какую-либо дополнительную информацию о возможных мотивах\шаблонах, имеющихся в найденном сете последовательностей. Эти шаблоны будут регулярными выражениями, генерированными при помощи программы Pratt.

в) Оценить эти сгенерированные шаблоны\паттерны при помощи поиска по сету последовательностей цитохрома P450 и сету последовательностей не-P450. Оценить их описательную и дискриминантную силу.

1. Проверьте, что вышеуказанный мотив (FMFEGHDTTA) присутствует во всех последовательностях набора изученных цитохромов P450 (в конце лабораторной работы). Вы можете использовать приложение patmatdb локально (предварительно установив пакет EMBOSS) или в любой из многочисленных её online реализаций - например <http://bips.u-strasbg.fr/EMBOSS/>.

Помощь, документация на patmatdb -

<http://emboss.sourceforge.net/apps/release/5.0/emboss/apps/patmatdb.html>.

2. Используя ScanProsite - <http://ca.expasy.org/tools/scanprosite/> - (Поиск «Motif(s) to scan for:» - правое поле), проведите поиск мотива FMFEGHDTTA среди последовательностей базы SwissProt (галочка - только около

UniProtKB/Swiss-Prot).

3. Загрузите найденные последовательности в формате Fasta - последовательно откройте каждую найденную последовательность и нажмите “fasta”. Создайте затем файл multi-fasta - все последовательности в одном файле.

4. Проверьте, что найденные последовательности специфичны в отношении искомого мотива - поместите их в поле Sequences to be scanned.

Теперь нам нужно попытаться автоматически генерировать мотивы, определяющие (характеризующие) набор полученных последовательностей белков P450, а затем протестировать качество этих шаблонов:

5. Используйте PRATT (<http://www.ebi.ac.uk/Tools/pratt/>) - это приложение, которое автоматически составляет регулярные выражения, описывающие эти последовательности. Пример работы PRATT -

<http://www.iu.uib.no/~inge/papers/mdl/small.html>.

- Произведите обратный поиск по базе SwissProt с использованием сгенерированных паттернов. Какие последовательности были найдены?

Эти поиски вернули исходные последовательности?

Какие иные последовательности (если есть) были найдены по введенным шаблонам? Являются ли они последовательностями P450?

- Вы можете также использовать приложение IBM TEIRESIAS

<http://cbcsrv.watson.ibm.com/Tspd.html> для генерации

паттернов\шаблонов.

6. Используйте набор последовательностей P450, который вы сумели идентифицировать при помощи FMFEGHDTTA в качестве набора positive examples.

7. Случайным образом разделите этот набор на обучающую и тестовую выборки (в соотношении обучающая\тестовая между 1:1 и 1:3).

8. Используйте PRATT для обучения (получения) надёжного паттерна по обучающей выборке.

9. Произведите поиск в тестовом наборе (или в полной базе данных) по найденному паттерну\шаблону. Вы можете использовать приложение EMBOSS patmatdb (для поиска с использованием паттернов, предварительно найденных в обучающем наборе. Документация на patmatdb -<http://emboss.bioinformatics.nl/cgi-bin/emboss/>).

10. Создайте новый тестовый набор с negative examples в fasta- формате - возьмите последовательности глобинов от разных организмов (см. предыдущие лаб. работы или Приложение). Тут мы предполагаем, что последовательности глобинов не содержат мотив FMFEGHDTTA. Является ли это предположение верным? Вы можете проверить это предположение, поместив эти последовательности в поле поиска Sequences to be scanned на странице поиска ScanProsite, а затем произведите поиск в базе данных Prosite для того, чтобы увидеть мотивы, соответствующие последовательностям глобинов.

11. Теперь вы имеете два тестовых набора (позитивный тест-набор из п.6 и негативный тест-набор с данными о глобинах).

12. Используйте приложение EMBOSS patmatdb для поиска некоторых паттернов\мотивов, полученных ранее благодаря PRATT, в позитивном и негативном тест-наборах последовательностей.

13. Вы должны знать, что в каждом поиске True Positives (TP), True Negatives (TN), False Positives (FP) и False Negatives (FN) значат следующее:

- TP - шаблон найден в позитивном наборе;
- FN - шаблон не найден в позитивном наборе;
- FP - шаблон найден в негативном наборе;
- TN - шаблон не найден в негативном наборе.

14. Используйте ваши результаты для подсчета различных показателей качества для шаблона (см. rating_patterns.pdf). Убедитесь в том, что когда вы выбираете слабый шаблон, предложенный PRATT, то и показатели качества будут плохими. Но что является интуитивным показателем слабого шаблона?

Можете ли вы сказать что-либо, оценить силу паттерна только глядя на него вне связи с целевыми последовательностями? Всегда ли короткие паттерны являются слабыми?

15. Проведите обратную проверку - создайте шаблоны-паттерны для глобинов (теперь они выступают в качестве позитивного набора). Используйте набор P450 в качестве негативного. Удалось ли вам обнаружить дискриминирующие, определяющие шаблоны для глобинов?

Другие программы и ресурсы

а) PPSEARCH (<http://www.ebi.ac.uk/Tools/ppsearch/>): производит поиск мотивов в исследуемой последовательности, быстро сравнивает введенную последовательность с шаблонами\паттернами,

накопленными в базе данных шаблонов Prosite и пытается предсказать функцию неизвестного белка. Приложение запрашивает на вход последовательность белка. ДНК\РНК могут быть транслированы в последовательность белка при помощи transeq

(<http://www.ebi.ac.uk/Tools/emboss/>).

б) База данных EMOTIF (<http://motif.stanford.edu/> или <https://bioinformatics.cs.vt.edu/emotif/>) - это коллекция более чем 170000 высокоспецифичных и чувствительных мотивов последовательностей белка, представляющих консервативные свойства и биологические функции. Эти белковые мотивы были выделены при помощи более чем 7600 выравниваний последовательностей в базе данных BLOCKS+ (23 июня 2000) и всех (8244) выравниваний белковых последовательностей в базе PRINTS. Для построения использовался также алгоритм emotifmaker, разработанный Nevill-Manning с соавторами. Благодаря тому, что аминокислоты и их группы в найденных мотивах отражают критические позиции в белках, закрепленных эволюционно, поисковые алгоритмы, примененные в EMOTIF, могут идентифицировать и классифицировать весьма большую группу различных белковых последовательностей - большую, чем методы, основанные на глобальном выравнивании последовательностей. База данных

мотивов\паттернов EMOTIF доступна по адресу <http://motif.stanford.edu/>

и <https://bioinformatics.cs.vt.edu/emotif/>.

- Emotif Maker (<https://bioinformatics.cs.vt.edu/emotif/emotif-maker.html>) - программа для генерации мотива\шаблона (регулярных выражений для последовательностей,

соответствующих функционально важным областям) на основании результатов выравнивания;

- Emotif Scan (<https://bioinformatics.cs.vt.edu/emotif/emotif-scan.html>)

- программа для поиска последовательности, соответствующей созданному регулярному выражению, в базе данных последовательностей.

- Emotif Search (<https://bioinformatics.cs.vt.edu/emotif/emotif-search.html>)

- программа для сопоставления последовательности определенным участкам по базе данных блоков (blocks (http://blocks.fhcrc.org/blocks/help/about_blocks.html)) и отпечатков (prints (<http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/index.php>)).

Блоки - это сегменты без блоков в множественном выравнивании последовательностей, соответствующие высоко консервативным регионам (highly conserved regions) в белках. Отпечатки - это резюме белковых «отпечатков пальцев» или fingerprints - группа консервативных мотивов, используемая для описания, аннотирования, характеристики семейств белков.

в) PHI-BLAST

(http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?PROGRAM=blastp&BLAST_PROGRAMS=blastp&PAGE_TYPE=BlastSearch&SHOW_DEFAULTS=on&LINK_LOC=blasthome

или <http://mobylyle.pasteur.fr/cgi-bin/portal.py?#forms::phiblast>) - это BLAST, использующий паттерны

определенного формата (формат похож на Prosite, см. сайт PHI-BLAST) для инициации поиска с использованием BLAST. Попробуйте провести поиски с использованием PHI-BLAST.

Контрольные вопросы

1. Предсказание третичной структуры белка. Распознавание фолда.

ПО, сервисы.

2. Динамическое программирование и выравнивание

3. последовательностей. Способы оптимизации поиска - FASTA, BLAST

4. Классификации белков. Базы данных Pfam, SCOPE, CATH

5. NCBI, ENTREZ и BLAST - назначение, инструменты, задачи.

6. Штрафы за вставку промежутка, схемы, различия.

7. Профиль и консенсус. Сходство и различия.

Приложение к лабораторной работе № 5

Последовательности и материалы, необходимые для выполнения лабораторной работы

Цитохромы P450:

>1cpt00

>1bvyA0

>1eurA0

>1akd00

>1cmnA0

Globin Sequences (Beta-Chain)

Последовательности глобинов для лабораторной работы № 5 аналогичны таковым для лабораторной работы № 4.

ЛАБОРАТОРНАЯ РАБОТА № 6 ФИЛОГЕНЕТИЧЕСКИЕ ДЕРЕВЬЯ

Цель лабораторной работы: получение практических навыков и понимания филогенетических деревьев и их построения.

Выполнение работы

Программное обеспечение

Polytree: Вы можете использовать web-версию программы по адресу

<http://www.dina.dk/~sestoft/bsa/Match7Applet.html>

Phylip: Онлайн доступ к сервису Phylip:

<http://bioweb2.pasteur.fr/phylogeny/intro-en.html>. Список программ,

входящих в пакет Phylip, находится по адресу <http://evolution.genetics.washington.edu/phylip/programs.html>. С сервера Phylip можно скачать

(<http://evolution.genetics.washington.edu/phylip/getme.html>) и установить этот пакет для использования на локальном компьютере. Там же находятся и исходные коды пакета, инструкции и примеры и ссылки на другие программы для филогенетических исследований.

BLAST: <http://blast.ncbi.nlm.nih.gov/Blast.cgi> и его разновидность PSI-Blast (меню distance tree или results).

Упражнения

А. Построение филогении в семействе глобинов (бета-цепи) Используйте ClustalW/Phylip для вычисления дистанций и матриц замен. а) Последовательности глобинов в формате fasta, необходимые для проведения работы, приведены в приложении.

б) Откройте любую из реализаций ClustalW (онлайн версии, локальные реализации или в EMBOSS (программа emma, которая является оболочкой ClustalW)).

- Загрузите файл fasta, содержащий последовательности глобинов, при помощи кнопки Browse.

- Оставьте настройки, установленные по умолчанию. Формат вывода

должен быть установлен в PHYLIP.

- Нажмите Submit - алгоритм выравнивания будет запущен.

- Скопируйте выравнивание в новый текстовый файл и сохраните его (это будет служить входными данными для программ Phylip).

в) Откройте каталог, в котором вы сохранили файл выравнивания в формате Phylip.

г) С помощью программы protdist из пакета Phylip (или онлайн версии:

<http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::protdist> и

<http://www.cbib.u-bordeaux2.fr/pise/protdist.html>) рассчитайте матрицу расстояний, соответствующую построенному выравниванию глобинов. Введите имя сохранённого выравнивания в формате Phylip в окно protdist.

д) Сохраните результаты работы protdist в файле matrix.

е) Теперь мы можем построить деревья по алгоритму neighbor-joining, используя программу neighbor из пакета Phylip или на сервере <http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::neighbor>. Введите матрицу расстояний в окно ввода. Заметьте, что на данный сервис позволяет выбрать метод построения дерева также и по алгоритму UPGMA. Вы можете попробовать построить деревья по обоим алгоритмам и сравнить их.

ж) Neighbor выводит два различных файла - outfile (визуализацию дерева) и treefile (расстояния).

з) Вы можете построить неукорененное дерево для neighbor joining , используя программу drawtree или укорененное (для UPGMA) при помощи drawgram. Пригодится созданный treefile.

и) Вы также можете построить филогенетические деревья, используя программы fitch или kitsch из пакета Phylip (или онлайн версии <http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::fitch> и <http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::kitsch>). Для визуализации используйте уже известные drawtree или drawdram.

Можно пройти более простым автоматизированным путем, поместив последовательности в поле ввода на сервере <http://mobyli.pasteur.fr/cgi->

[bin/portal.py?#forms::protein distance phylogeny](http://bin/portal.py?#forms::protein%20distance%20phylogeny)

В. Начало и эволюция ВИЧ

Данные:

Нуклеотидные последовательности ВИЧ в формате fasta (в конце лаб. работы). Эти последовательности ВИЧ подтипа С из Южной Африки (последовательности C.ZA) и последовательности из Индии (C.IN) или

Fasta-файлы, содержащие аминокислотные последовательности для envelope, gag и pol белков изолятов, приведенных ниже (см. последовательности в fasta в приложении и

<http://artedi.ebc.uu.se/course/UGSBR/hiv/>):

Таблица 3.

no	Isolate	Accession no (GenBank, ENTREZ)	Subtype/animal
1	HIV-1 ^Δ	K03454 http://www.ncbi.nlm.nih.gov/nucore/K03454	D
2	HIV-1 _M	K02013 http://www.ncbi.nlm.nih.gov/nucore/K02013	B
3	HIV-W	M27323 http://www.ncbi.nlm.nih.gov/nucore/M27323	D
4	HIV-2 _{D205}	X61240 http://www.ncbi.nlm.nih.gov/nucore/X61240	B
5	HIV-2 ^Δ _D	M15390 http://www.ncbi.nlm.nih.gov/nucore/M15390	A
6	HIV-2 _{v/-}	M31113 http://www.ncbi.nlm.nih.gov/nucore/M31113	A
7	HIV-2 _π	L07625 http://www.ncbi.nlm.nih.gov/nucore/L07625	B
8	SIV _{mac}	M19499 http://www.ncbi.nlm.nih.gov/nucore/M19499	macaque
9	SIV _{ch}	X52154 http://www.ncbi.nlm.nih.gov/nucore/X52154	chimpanzee

10	SIV v agm	M58410 http://www.ncbi.nlm.nih.gov/nuccore/M58410	African green monkey
----	--------------	---	-------------------------

11	SIV v man	X14307 http://www.ncbi.nlm.nih.gov/nucore/X14307	mangabey
----	--------------	---	----------

а) Проведите филогенетическое исследование этих последовательностей (MSA по ClustalW с выводом в Phylip), аналогичное проделанному в п. А.

б) Используя подходящие приложения из пакета Phylip:

- Постройте матрицу расстояний с помощью dnadist (почему?);
- Постройте дерево;
- Визуализируйте построенное в п.Б филогенетическое дерево.

С. Начало и эволюция H5N1

Самостоятельно постройте филогенетические деревья по алгоритму UPGMA и Neighbor Joining для различных последовательностей ДНК птичьего гриппа (см. приложение). Объясните различия в построенных деревьях.

D. Метод максимальной бережливости Maximum Parsimony (MP)

В этом упражнении вы будете реконструировать филогенетическое дерево, используя программу Dnapars из пакета Phylip. Как и в предыдущих упражнениях, вы можете установить Phylip на компьютер и использовать это ПО локально (<http://evolution.genetics.washington.edu/phylip/install.html>) или воспользоваться онлайн сервисом, предоставляющим функционал Phylip (<http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::dnaps>).

Справочная информация по Dnapars -
<http://evolution.genetics.washington.edu/phylip/doc/dnaps.html>. Важно:

Dnapars используется только для реконструкции

филогенетических деревьев по методу MP для последовательностей ДНК. Для работы с последовательностями белков нужно использовать программу Protpars (<http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::protpars>).

а) Установите Phylip на компьютер или откройте <http://mobyli.pasteur.fr/cgi-bin/portal.py?#forms::dnaps> (дальнейшие шаги приведены для случая использования онлайн-сервиса).

б) Постройте выравнивание последовательностей из файла cyt.b.txt (см.

приложение) и сохраните его в формате, поддерживаемом Phyliр. в)

Введите выровненные последовательности в окно ввода.

г) Оставьте параметры по умолчанию и нажмите RUN. Введите действующий адрес электронной почты.

д) Посмотрите на построенное дерево. Какие данные вывел сервис? Проанализируйте и объясните их. Сохраните дерево и сопутствующие результаты.

е) Постройте дерево последовательностей из cytb.txt (приложение), используя Phyliр и алгоритм NJ. Сохраните дерево. Сравните филогенетические деревья.

ж) В поле Tree file выберите view with archaeopteryx. Сравните построенные деревья. Похоже ли это дерево на полученное при помощи ClustalW?

з) Bootstrap анализ.

- Откройте ClustalW (<http://clustalw.ddbj.nig.ac.jp/>)

- Поместите в окно ввода последовательности из cytb.txt (приложение)

- В поле Bootstrap отметьте «ON».

- Нажмите «Submit».

- Сохраните на диске файл выравнивания *.aln.

- Выберите в пакете Phyliр раздел Phylogeny - tree analyser и проанализируйте построенное выравнивание и bootstrap значения.

Контрольные вопросы

1. Выравнивание и его статистическая достоверность. Bootstrap.
2. Докинг - цель и задачи. Трудности.
3. Жёсткий докинг. Методы, применение.
4. Гибкий докинг.
5. Экспериментальное определение структур белка. Рентгеноструктурный анализ.
6. Экспериментальное определение структур белка. Ядерно-магнитный резонанс.

7. Экспериментальное определение структуры белка. Оценка качества полученной структуры.

Приложения к лабораторной работе № 6:

Globin Sequences (Beta-Chain)

Для выполнения лабораторной работы используйте последовательности глобинов из лабораторной работы № 4.

Последовательности HIV (к п. В):

>C.ZA_AY047297

>C.ZA_AY047296

>C.ZA_AY047295

>C.ZA_AY047299

>C.ZA_AY047298

>C.IN_Y17891

>C.IN_Y17892

>C.IN_Y17893

>C.IN_Y18199

HIV envelope proteins (к п. В):

>H1_ELI_D

>H1_LAI_B

>H1_MAL

>H1_NDK_D

>H2_D205_B

>H2_ROD_A

>H2_ST_A

>H2_UCI_B

>S_MAC

>S_CP2

>S_AGM

>S_MAN

HIV gag proteins (к п. В):

>H1_ELI_D

>H1_LAI_B

>H1_MAL

>H1_NDK_D

>H2_D205_B

>H2_ROD_A

>H2_ST_A

>H2_UCI_B

>S_MAC

>S_CP2

>S_AGM

>S_MAN

HIV pol proteins (к п. B):

>H1_ELI_D

>H1_LAI_B

>H1_MAL

>H1_NDK_D

>H2_D205_B

>H2_ROD_A

>H2_ST_A

>H2_UCI_B

>S_MAC

>S_CPZ

>S_AGM

>S_MAN

ДНК вирусов H5N1 (к п. C):

Для выполнения лабораторной работы используйте последовательности ДНК вирусов птичьего гриппа из лабораторной работы № 4.

Последовательности к п D.

>Papio_hamadryas (baboon)

- >Pan_troglodytes (chimpanzee)
- >Homo_sapiens (human)
- >Pongo_pygmaeus (orangutan)
- >Pan_paniscus (pygmy chimpanzee)
- >Phoca_vitulina (harbor seal)
- >Gorilla_gorilla (gorilla)

ЛАБОРАТОРНАЯ РАБОТА № 7 «ТЕКСТОВЫЕ МЕТОДЫ В БИОИН- ФОРМАТИКЕ»

Цель работы: продемонстрировать потребности и пользу от методов text mining (TM) в системной биологии и биоинформатике.

Теоретические выкладки

TM - это метод извлечения информации из больших объемов текста. Очевидно, что эта информация зависит, в том числе, и от потребностей пользователя, от правильности составления им запросов. В системной биологии одной из целей является представление организма в виде схем и диаграмм. Иными словами организм должен быть представлен как система взаимосвязей их базовых компонентов, таких, как гены и белки. Многие из таких взаимосвязей могут наблюдаться исследователем независимо от других исследовательских групп в мире. Некоторые взаимосвязи могут являться побочным продуктом исследований, проводимых с другой целью, а не для установления случайно выявленной (или вообще какой либо) взаимосвязи.

На данный момент мы будем рассматривать белок-белковые взаимодействия. Белки являются продуктами экспрессии генов. Каждый белок выполняет специфическую, присущую ему функцию в клетке. Их можно рассматривать как «разумные» частицы, выполняющие свои задачи в клетках. Белки могут влиять друг на друга при связывании и, возможно, происходящем при этом ингибировании или активации. Это значит, что взаимодействующие белки могут образовывать сети или пути. Такие пути могут быть классифицированы, разделены на метаболические, сигнальные, транскрипционные и другие.

В первой части работы мы проведем исследование одного сигнального пути. Этот путь является небольшой частью большей сети путей, вовлеченной в регуляцию деления клетки. Когда этот сигнальный путь ломается, распадается, то клетка начинает делиться бесконтрольно, что, в свою очередь, приводит к раку. Поэтому этот путь является исключительно важным для

исследователей механизмов канцерогенеза. Мы будем использовать доступные инструменты (медицинские базы данных) для изучения различных аспектов этого пути.

Во второй части работы мы уделим внимание известному инструменту TM - GATE (<http://gate.ac.uk/>). Мы будем использовать его (расчет F-score) для оценки распознавания определенных терминов в биологических текстах. GATE является довольно мощным инструментом, однако он может быть неточен при работе с биологическими текстами. Мы проверим насколько эффективно инструмент общего предназначения работает с медико-биологическими текстами и насколько сильно биологические тексты отличаются от, скажем, новостных текстов.

Выполнение работы

Инструменты и ПО, используемое в лабораторной работе:

The Gene Ontology (<http://www.geneontology.org/>) - иерархический курируемый словарь, описывающий «продукты экспрессии генов в терминах их ассоциаций с биологическими процессами, клеточными компонентами и молекулярными функциями без учета видоспецифичности». Этот инструмент поиска может быть полезен при изучении компонентов различных путей.

KEGG (http://www.genome.jp/dbget-bin/show_pathway?hsa04010+5894) - сайт предназначен для представления и визуализации различных метаболических, сигнальных и иных путей. Сайт содержит ссылки на аннотации и описания на все белки, вовлеченные в метаболические пути и сети, а также перекрестные ссылки на пути, содержащие общие белки и ссылки на белки в других базах данных.

Entrez Gene (<http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=search&db=gene>)
- база

данных, содержащая информацию о белках и генах. Она также имеет ссылки на оригинальные статьи, имеющие отношение к имеющимся записям о последовательностях. В Entrez Gene хранится информация о синонимах для всех имеющихся в ней белках - это помогает составлять более точные и

полные запросы к базе.

PubMed (<http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=search&db=pubmed>) - главный портал для доступа к базам данных Medline. Эта база данных содержит цитаты на статьи в рецензируемых журналах (статьи, публикуемые в журналах с расширенным рецензированием). Цитаты, хранящиеся в PubMed, имеют полную библиографическую информацию и аннотации для большинства научных статей.

Web of Knowledge (<http://portal.isiknowledge.com/?DestApp=WOS>) - поисковая машина для медико-биологических текстов и статей. Она находится в собственности и может быть доступна только из университетских сетей или по подписке.

EBIMed (<http://www.ebi.ac.uk/Rebholz-srv/ebimed/index.jsp>) -поисковая машина, которая представляет результаты поиска в виде таблицы корреляций, терминов и гиперссылок, встречающихся в статьях. Мы будем использовать ее для проверок связей, описанных в путях. Эта поисковая машина имеет в своем составе элементы ТМ, такие, как распознавание именованных сущностей с использованием лексикона.

ChiliBot (<http://www.chilibot.net/>) - еще одна поисковая машина, которая несет в себе элементы ТМ. Это позволяет пользователям проводить поиск по нескольким ключевым словам и затем выводить результаты по каждому из них или по всем сразу и строить сети одновременного появления этих ключевых слов в предложениях - сети взаимодействия и взаимосвязей терминов.

GATE (<http://gate.ac.uk/>) - мощное средство ТМ, которое позволяет вам использовать предварительно заготовленные компоненты (такие, как блоки) для построения вашей собственной ТМ машины. GATE была разработана для работы с текстами общего назначения. Инструмент недостаточно пригоден для поиска по биологическим текстам, так как для их обработки требуются дополнительные компоненты. В этих случаях нужно использовать инструмент Annotation Diff для оценки работы on-line версии программы на выборке из

аннотированной вручную базы GENIA.

Lingpipe (<http://alias-i.com/lingpipe/web/demos.html>) - онлайн демоверсия ТМ Java библиотеки, которая располагает компонентами, обученными на биологических текстах. Однако в ней отсутствуют хороший интерфейс пользователя и некоторые другие средства и инструменты для оценки.

Сигнальный путь RKIP, показанный на рисунке 3, очень хорошо описан в статье Cho et al (2003)¹. Понимание механизмов, заложенных в этом сигнальном пути исключительно важно при изучении рака. Исследователи проводят десятки экспериментов на клетках, в которых имеется RKIP, удаляя или блокируя части пути, стимулируя рецепторы белки, которые инициализируют каскады реакций или добавляя различные соединения (лекарства). Структура и функция RKIP по-прежнему стали проявляться благодаря многочисленным экспериментам. Путь RKIP состоит из части MAPK-пути, который содержит белок Raf-1, взаимодействующий с ингибитором Raf-киназой (RKIP).

Представлены белки Raf-1, RKIP, ERK-PP и другие, образующие этот путь. Raf-1/RKIP представляют их взаимодействие, тогда как «PP», находящийся после MEK или ERK (MEK-PP и ERK-PP соответственно) обозначает активированную форму белка. Это, в свою очередь, означает, что ERK-PP не идентичен ERK, однако не существует стандартной формы записи этих названий, и поэтому многие ученые употребляют в написании pERK или ERKPP.

Как было сказано выше, в первой части лабораторной работы мы будем использовать онлайн-ресурсы для ознакомления с различными аспектами этого сигнального пути и для более полного понимания тех трудностей, с которыми сталкиваются биологи при попытке получения более специфичной биологической информации при помощи стандартных инструментов.

¹ Cho, K.-H., et al., Mathematical Modeling of the Influence of RKIP on the ERK Signaling Pathway, in Proceedings of the First International Workshop on Computational Methods in Systems Biology. 2003, Springer-Verlag. p. 127-141.

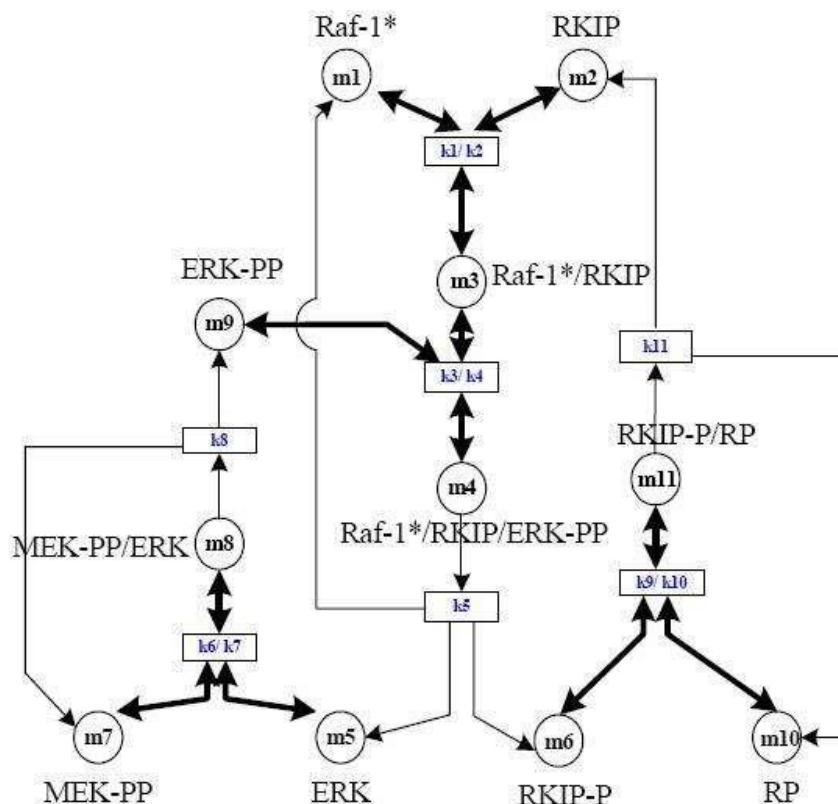


Рисунок 3. Сигнальный путь RKIP.

Часть 1

Некоторые из этих ресурсов не обладают большой скоростью, в особенности Gene Ontology AmiGO, поэтому стоит выполнять задания лабораторной работы параллельно в разных окнах браузера.

1. Знакомство с сигнальным путём RKIP

В этом упражнении вы будете находить различные компоненты пути и определять их синонимы. Вы попытаетесь научиться понимать функции путей при помощи анализа составляющих их компонентов и проходам по сигнальным путям и соединениям, связывающим узлы путей.

а) Откройте сайт KEGG http://www.genome.jp/dbget-bin/show_pathway?hsa04010+5894. Взгляните на полный MAPK путь. Можете ли вы сказать, где конкретно путь RKIP вписывается в большую сеть MAPK? Какие компоненты RKIP являются также компонентами MAPK? Кликните на любом из компонентов MAPK чтобы увидеть дополнительную информацию по выбранному компоненту. Какие ещё сигнальные пути имеют

в своем составе выбранный вами компонент\соединение?

б) Откройте в другом окне браузера для поиска RKIP сайт Entrez Gene <http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=search&db=gene>. В списке «Search» выберите «Protein». Что вам удалось узнать об этом белке? Выпишите эту информацию и его синонимы.

в) Более подробный поиск данных об этом белке можно провести на сайте The Gene Ontology (работа с ним иногда занимает много времени) <http://www.geneontology.org/>.

2. Изучение сигнального пути Используйте базы данных PubMed

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=pubmed> или Web of Knowledge

<http://portal.isiknowledge.com/portal.cgi?DestApp=WOS&Func=Frame> для поиска информации о RKIP пути. Попробуйте провести такой поиск с использованием различных ключевых слов, которые, по вашему мнению, могут помочь вам получить больше информации о компонентах сигнального пути из литературных источников.

3. Использование специализированных веб ресурсов

Откройте EBI Med (<http://www.ebi.ac.uk/Rebholz-srv/ebimed/index.jsp>) и попытайтесь поискать компоненты сигнального пути. Какие результаты возвратил вам сервер?

Выберите пять компонентов и произведите по ним поиск с использованием ChiliBot (<http://www.chilibot.net/>). Насколько полученная вами карта соответствует исходному сигнальному пути? Нажмите на кнопку «Edit Synonyms» (слева). Добавьте выбранные вами в п.1 синонимы. Произведите повторный поиск.

Часть 2

В этой части мы попытаемся произвести распознавание автоматически именovaných биологических сущностей с использованием средства Lingpipe и GATE. Вы можете получить дополнительную информацию и навыки работы с

GATE, просмотрев видео на странице инструмента ([http:// gate.ac.uk/demos/gate-tutorial-all/part-one-all-with-annic-and- ontology.html](http://gate.ac.uk/demos/gate-tutorial-all/part-one-all-with-annic-and-ontology.html)).

1. Для оценки эффективности различных программных средств очень важно иметь эталонный тестовый набор для тестирования. Этот набор обычно называется «Золотой стандарт». В этом упражнении мы будем использовать очень маленькую часть GENIA corpus (<http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/>), дополненную некоторыми деталями в целях выполнения этой работы.

2. Скачайте и установите GATE на ваши компьютеры <http://gate.ac.uk/download/>. На странице есть дистрибутивы для *NIX, Windows и MacOS систем.

3. Запустите GATE. Параметры окон и шрифтов настраиваются в Options Configuration.

4. Откройте меню File^New Language Resource^GATE document. В поле «имя» введите «genia» и затем нажмите на кнопку «browse» для открытия файла (текст из приложения 1 сохраните в отдельном файле .xml).

5. Вы увидите, что файл с Genia corpus откроется в Language Resources в левой панели. Двойной щелчок на ней открывает вторую закладку в центральном окне. Кликните на Annotation Sets, разверните исходный список и попробуйте выбирать различные закладки. Вы увидите, что текст подсвечивается по-разному. Наведение указателя мыши на подсвеченные части текста вызывает всплывающие окна. Проведите этот опыт. Что отображается в всплывающих окнах при наведении указателя мыши на подсвеченный текст?

6. Сохраните неразмеченный текст из приложения 2 в файле .txt. Зайдите на страницу Lingpipe <http://alias-i.com/lingpipe/web/demo-ne.html> и попытайтесь открыть страницу для поиска по тексту Named Entity Demo on the Web^English Biomedical Text: GENIA Corpus (TokenShapeChunker). Введите в открывшееся текстовое поле текст из приложения 2 (неразмеченный текст

Genia Corpus). Нажмите кнопку «Submit Text».

7. Сохраните результаты на диск и затем откройте сохраненный файл как документ GATE.

8. Внутри окна GATE нажмите Tools ^Annotation Diff. Там, где написано Document (слева вверху в окне Annotation Diff) выберите ваш документ genia в первом выпадающем меню и ваш результат lingpipe во втором. Установите Original Markup для обоих в Annotation Set. Кликните на Do Diff.

9. Обратите внимание на F-score и метки, присвоенные программой и имеющиеся в «Золотом стандарте». Посмотрите на частичные совпадения и то, как они соответствуют друг другу - вероятно, будут лишь незначительные различия в расстановке проделов в двух документах.

10. Сохраните текст из приложения 3 в файле .html.

11. Откройте File^New Language Resource^GATE corpus. Задайте приемлемое имя и нажмите на иконку со списком справа. Добавьте genia документ и документ с новостями.

12. Откройте File^Load ANNIE System^ With defaults.

13. Сделайте двойной щелчок по новому приложению ANNIE, которое отображается под Applications на левой панели. Убедитесь в том, что ваш Corpus документ выбран в выпадающем меню в центральной панели и нажмите Run.

14. Сделайте двойной щелчок на документе с новостями и разверните Annotation Sets в верхнем списке. Затем попробуйте выбирать различные метки, наблюдая за тем, какие аннотации выбирает GATE в этом простом новостном тексте.

15. Посмотрите на ваш документ genia и определите, какие аннотации были выбраны в нем GATE.

Приложения к лабораторной работе № 7 При-

ложение 1. GENIA corpus .xml Приложение 2

Genia corpus неразмеченный

MEDLINE:95369245

Экспрессия гена IL-2 и активация NF-каппа В через CD28 требует производства реактивного кислорода 5-липоксигеназой.

Активация поверхностного рецептора CD28 обеспечивает основной кости-мулирующий сигнал для активации Т-клеток, что приводит к усилению продукции интерлейкина-2 (IL-2) и пролиферации клеток.

В первичных Т-лимфоцитах мы показываем, что лигирование CD28 приводит к быстрому внутриклеточному образованию интермедиатов реактивного кислорода (ROI), которые необходимы для CD28-опосредованной активации NF-каппа В/CD28-чувствительного комплекса и экспрессии IL-2.

Было обнаружено, что разграничение сигнального каскада CD28 включает активность протеинтирозинкиназы с последующей активацией фосфолипазы A2 и 5-липоксигеназы.

Наши данные свидетельствуют о том, что метаболиты липоксигеназы активируют образование ROI, которые затем индуцируют экспрессию IL-2 посредством активации NF-каппа В.

Эти результаты должны быть полезны для терапевтических стратегий и разработки иммунодепрессантов, нацеленных на костимуляторный путь CD28.

СРЕДНЯЯ ЛИНИЯ: 95333264

Сайт пери-каппа В опосредует активацию энхансера вируса иммунодефицита человека типа 2 в моноцитах, но не в Т-клетках.

Вирус иммунодефицита человека типа 2 (ВИЧ-2), как и ВИЧ-1, вызывает СПИД и связан со случаями СПИДа в первую очередь в Западной Африке.

ВИЧ-1 и ВИЧ-2 обнаруживают значительные различия в последовательности нуклеиновых кислот и естественном течении клинического заболевания.

В соответствии с этими различиями мы ранее продемонстрировали, что энхансерная/промоторная область ВИЧ-2 функционирует совершенно иначе,

чем у ВИЧ-1.

В то время как активация энхансера ВИЧ-1 после стимуляции Т-клеток в основном опосредуется связыванием фактора транскрипции NF-каппа В с двумя соседними сайтами каппа В в длинном концевом повторе ВИЧ-1, активация энхансера ВИЧ-2 в моноцитах и Т-клетки зависят от четырех цис-действующих элементов: одного сайта каппа-В, двух сайтов связывания, богатых пурином, PuB1 и PuB2, и сайта pets.

Теперь мы идентифицировали новый цис-действующий элемент в энхансере ВИЧ-2, непосредственно перед сайтом каппа-В, обозначенный как пери-каппа-В.

Этот сайт консервативен среди изолятов ВИЧ-2 и близкородственного обезьяньего вируса иммунодефицита, и анализы трансфекции показывают, что этот сайт опосредует активацию энхансера ВИЧ-2 после стимуляции моноцитарных, но не Т-клеточных линий.

Это первое описание энхансерного элемента ВИЧ-2, который демонстрирует такую моноцитарную специфичность, и для ВИЧ-1 не было четко определено ни одного сопоставимого энхансерного элемента.

В то время как ядерный фактор (факторы) как моноцитов периферической крови, так и Т-клеток связывается с пери-каппа-В-сайтом, анализы сдвига электрофоретической подвижности предполагают, что либо другой белок связывается с этим сайтом в моноцитах, чем Т-клетки, или что белок, распознающий этот энхансерный элемент, подвергается дифференциальной модификации в моноцитах и Т-клетках, что подтверждает данные о трансфекции.

Кроме того, в то время как специфическое конститутивное связывание с сайтом пери-каппа В наблюдается в моноцитах, стимуляция эфирами форбола вызывает дополнительное специфическое связывание.

Понимание специфической для моноцитов функции фактора пери-каппа В может в конечном итоге дать представление о различной роли, которую моноциты и Т-клетки играют в патогенезе ВИЧ.

ЛАБОРАТОРНАЯ РАБОТА № 8 ФИЛОГЕНЕТИЧЕСКИЙ АНАЛИЗ

Цель работы - изучение возможностей горизонтального переноса генов у бактерий на примере т-РНК синтетазы.

Выполнение работы

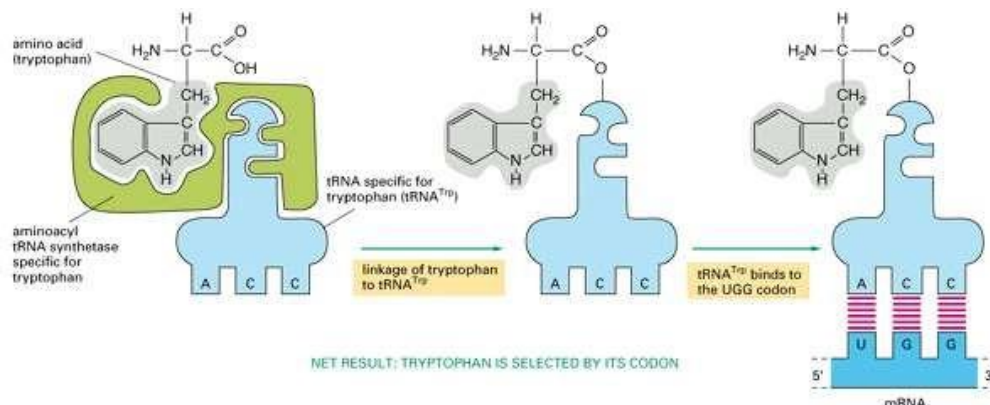


Рисунок 4. Работа т-РНК синтетазы.

Роль т-РНК синтетазы.

При помощи BLAST выберите последовательности бактериальных т-РНК. Постройте филогенетическое дерево, используя выбранные последовательности.

Постройте филогенетическое дерево видов, используя последовательность ssu-16s (последовательность рибосомальной РНК 16s субъединицы).

Сравните полученные деревья и объясните их сходство и различия.

Контрольные вопросы

1. Интерпретация результатов выравнивания - Score, E-value. Проверка достоверности полученного выравнивания.

2. BLAST. Алгоритм поиска оптимального выравнивания. Состав BLAST на NCBI.

3. По приведенной матрице расстояний постройте филогенетическое дерево (Neighbor Joining method, UPGMA). Опишите процесс построения.

	1	2	3	4
1	0	0.3	0.5	0.6

2	0	0.6	0.5
3		0	0.9
4			0

4. Штрафы за вставку промежутков. Аффинные и линейные штрафы, их сравнение и влияние на выравнивание.

5. Что может сказать об эволюционном процессе организмов свойство ультраметричности построенного для них дерева (при условии его истинности)?

6. Свойство аддитивности филогенетического дерева. Его определение.

ЛАБОРАТОРНАЯ РАБОТА №9 «ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ НА ЯЗЫКЕ PYTHON»

Цель работы: Проверить знания базовых инструментов, которые используются для обращения с данными в проектах по машинному обучению - библиотек numpy, pandas, средств визуализации matplotlib, seaborn.

Теоретические выкладки

Для успешного выполнения заданий и проектов по машинному обучению необходимо постоянное манипулирование данными в определенных форматах. Очень облегчают эту задачу специализированные библиотеки. В экосистеме Python почти всегда используются модули numpy и pandas для анализа и преобразования данных. Поэтому для эффективной работы с моделями машинного обучения необходимо виртуозно владеть этими базовыми библиотеками.

Среди базовых возможностей языка программирования Python при работе с анализом данных и машинным обучением чаще всего используются следующие:

- индексирование массивов и срезы
- генераторные выражения с условиями и циклами
- анонимные функции
- Среди средств библиотеки numpy чаще всего применяются

следующие инструменты:

- функции генерации массивов arange, linspace, logspace
- функция генерации сетки mgrid
- функции создания матриц diag, zeroes, ones
- создание случайных значений - пакет random
- форма массива - shape
- изменение формы массива - reshape
- оператор многоточия - ...

- сложное индексирование
- индексные маски
- поэлементные операции

Библиотека `pandas` особенно полезна для манипуляции двумерными массивами информации. К самым часто используемым ее возможностям в машинном обучении относятся следующие:

- сохранение и чтение файлов в формате `csv`, `xls`, `xlsx`
- аналитические функции `info`, `describe`, `head`
- итерация по строкам и столбцам, обращение к строкам и столбцам по

номерам и индексам/названиям

- объединения и соединения таблиц
- агрегатные функции и группировка
- замена отдельных значений в таблице в том числе по условию Кроме то-

го нам часто придется визуализировать данные в виде различных

графиков, гистограмм. Средства визуализации тоже очень важны и полезны. Аналитик должен владеть разными типами графиков и выбирать наиболее подходящий для решения конкретной задачи. Помните, что визуализация нужна для того, чтобы сделать видимыми какие-то скрытые зависимости в данных. То есть визуализация должна быть простой, наглядной и понятной, с явными целями и выводами.

Эту и все последующие работы следует выполнять в ноутбучном редакторе `Python`, который позволяет выполнять код по ячейкам. Такой формат работы больше подходит для прототипирования и построения проекта по анализу данных и машинному обучению. Рекомендуется использовать `Jupyter notebook` или `Google Colab`.

Задания для самостоятельного выполнения

1. С помощью массивов `numpy` создайте таблицу умножения.
2. Создайте функцию, которая принимает как аргументы целое число `N` и первый элемент (вещественное число), и разность (вещественное число) и

создает матрицу `numpy` по диагонали, которой располагаются первые N членов арифметической прогрессии.

3. Сгенерируйте средствами `numpy` матрицу A 5 на 5, содержащую последовательные числа от 1 до 25. Используя срезы извлеките в плоский массив все нечетные элементы этой матрицы.

4. Создайте двумерный массив, содержащий единицы на границе и нули внутри.

5. Создайте две матрицы размером (5,5). Одна матрица содержит 5 в шахматном порядке как в задаче домашнего задания, другая имеет треугольную форму содержащую 5 на основной диагонали и в позициях выше ее, а ниже все 0. Посчитайте их детерминант и найдите обратные матрицы.

6. С помощью `pandas` загрузите датасет для предсказания цены квартиры, прилагающийся к этой работе.

7. Выведите на экран несколько первых и несколько последних строк файла.

8. Выведите с помощью методов `pandas` основные количественные параметры датасета: количество строк и столбцов, тип данных каждого поля, количество значений в каждом столбце, шкала измерения каждого численного поля.

9. Удалите из таблицы столбцы, содержащие идентификаторы, переименуйте все оставшиеся названия колонок на русском языке.

10. Выведите отдельно столбец, содержащий цену, по номеру и названию. Выведите первую, десятую и предпоследнюю строку таблицы по номеру и по индексу.

11. Выделите в отдельную таблицу последние десять строк. Уберите в ней столбец с ценой. Склейте ее с первоначальной таблицей при помощи `append`. Заполните отсутствующие значения цены средним по таблице.

12. Выделите пять последних колонок в отдельную таблицу. Удалите в ней строки, в которых цена ниже среднего. Присоедините эту таблицу к изначальной (выберите самый подходящий тип соединения).

13. Выведите таблицу, содержащую среднюю цену и количество квартир на каждом этаже из первоначального набора данных.

14. Сохраните получившуюся таблицу в файлы формата csv и xlsx. Прочитайте их и убедитесь, что данные отображаются корректно.

15. Создайте в Excel или другом табличном редакторе таблицу, содержащую несколько численных и текстовых полей. Прочитайте ее в программу при помощи pandas.

16. Дальнейшие задания производите используя изначальную версию датасета. Должны быть подписаны названия графиков, названия осей, указаны значения на осях. Оцениваться будет использование количества различных атрибутов при построении графиков и визуальная красота.

17. Постройте круговую диаграмму для признака Rooms, иллюстрирующую количество квартир в процентах в зависимости от количества комнат. Сделайте сектор с наибольшим числом квартир выдвинутым.

18. Постройте гистограмму по целевой переменной Price. Оцените визуально, по какой цене продаётся наибольшее количество квартир.

19. Постройте диаграммы рассеяния для признаков Rooms, Square, HouseFloor, HouseYear в зависимости от целевой переменной Price в одной области figure. Оцените визуально, есть ли среди них такие, на которых разброс точек близок к линейной функции.

20. Постройте ядерную оценку плотности целевой переменной Price. Оцените визуально, напоминает ли полученный график нормальное распределение. Постройте двумерную ядерную оценку плотности для целевой переменной Price и признака HouseFloor, затем оцените визуально на каких этажах и по какой цене продаётся основная масса квартир.

21. Постройте ящиковую диаграмму признака Square. Оцените визуально имеются ли выбросы, и, если да, то начиная с какого размера площади значение признака можно считать выбросом.

22. При помощи сетки графиков PairGrid визуализируйте попарные

отношения признаков Rooms, Square, HouseFloor, HouseYear, Price следующим образом: на диагонали - гистограммы, под диагональю - ядерные оценки плотности, над диагональю - диаграммы рассеяния. По результатам визуализации сделайте выводы.

23. Постройте тепловую карту матрицы корреляции (`df.corr()`) признаков Rooms, Square, HouseFloor, HouseYear, Price. По ней определите, какие признаки являются зависимыми (у таких признаков коэффициент корреляции близок к единице).

Контрольные вопросы

1. Какие структуры данных используются в Numpy? В чем их отличие от списков Python?
2. Какие функции для генерации массивов использует Numpy?
3. Какие способы предлагает Numpy для извлечения данных из массивов?
4. Что такое векторизация кода и почему это ускоряет работу программ?
5. Какие виды матричных операций реализованы в Numpy?
6. Какие функции используются для преобразования формы, размера и соединения массивов?
7. Какие две главные структуры данных используются в pandas? В чем их отличие?
8. Как происходит объединение двух таблиц в pandas?
9. Зачем нужны и как работают индексы в pandas.
10. Построение каких основных видов графиков используется при анализе данных в машинном обучении?
11. В чём разница между библиотеками matplotlib и seaborn? Каковы преимущества каждой из них?
12. Как задать размер графика в matplotlib?
13. Как установить стили в seaborn?
14. Для чего используют подграфики subplots?

15. Какие основные типы графиков реализованы в matplotlib? Что изображается на ящичковой диаграмме?

16. Как поменять палитру цветов у тепловой карты?

ЛАБОРАТОРНАЯ РАБОТА №10 «ОСНОВЫ РАБОТЫ С PANDAS»

Цель работы

На практике ознакомиться с основными приемами обработки данных с использованием библиотеки pandas, освоить основные понятия этой библиотеки

Методические указания

Для успешного выполнения данной лабораторной работы необходимо заранее ознакомиться с официальной документацией библиотеки pandas. Работу следует выполнять в интерактивной среде программирования Jupyter notebook. Оценивается полнота выполнения задания, знание и умение применять библиотечные функции

К данной работе прилагается файл с набором данных. Данный набор данных используется для открытого соревнования по машинному обучению Kaggle. Набор описывает основные характеристики пассажиров Титаника с информацией о том, выжил данный пассажир или нет. Часть полей носит категориальный, часть численный характер. Часть данных отсутствует.

Задания к выполнению

1. Получить файл набора данных, прилагающийся к этой работе.
2. Считать файл в pandas DataFrame
3. Ознакомиться с основными параметрами датасета, используя функции библиотеки pandas, включая следующие данные:
 1. Количество точек данных (измерений) в наборе
 2. Количество полей данных (атрибутов)
 3. Тип данных каждого поля, шкала каждого поля
 4. Количество отсутствующих значений для каждого поля
 5. Количество и названия константных и уникальных полей
4. Удалить из набора поля, не несущие информативной нагрузки - содержащие уникальные либо константные значения

5. Заменить возраст пассажиров на год рождения
6. Переименовать все названия колонок на русском языке
7. Заменить все отсутствующие данные средним значением (в случае числового поля) либо самым распространенным в категориальных полях.
8. Сохранить получившийся набор данных в формате CSV
9. Открыть данный файл в программе Microsoft Excel и сохранить в формате .xlsx
10. Считать файл .xlsx в новый DataFrame

Контрольные вопросы

1. Какие две главные структуры данных используются в pandas? В чем их отличие?
2. Назовите основные функции чтения данных из файлов
3. Напишите функцию записи набора данных A в файл с разделителями-запятыми
4. Как называется функция, выводящая на экран первые N строчек набора данных?
5. С помощью какой функции можно удалить поле набора данных? А строку?
6. Напишите участок кода, демонстрирующий возможность добавления новой строки к набору данных.
7. Напишите участок кода, демонстрирующий возможность добавления новой колонки (поля) к набору данных.

Темы для самостоятельного изучения

1. Объединение наборов данных в pandas.
1.[https://pandas.pydata.org/pandas-docs/stable/comparison_with_sql.html](https://pandas.pydata.org/pandas-docs/stable/comparison_with_sql.html)

2.<https://pandas.pydata.org/pandas-docs/stable/merging.html>

2. Работа с индексами

3. Группировка данных

3.<https://pandas.pydata.org/pandas-docs/stable/groupby.html>

4. Работа с датой-временем

4. <http://strftime.org/>

ЛАБОРАТОРНАЯ РАБОТА №11 ПРОСТОЙ АНАЛИЗ ДАННЫХ

Цель работы

Освоить основные приемы работы с библиотекой numpy.

Задания для выполнения

Вам даны данные результатов ЕГЭ по различным предметам. Выполните следующие действия:

1. Загрузите данные в ноутбук
2. Сделайте описательную статистику по набранному баллу
3. Найдите процент учащихся, выполнивших работу ниже среднего
4. Найти процент учащихся не сдавших экзамен
5. Постройте круговую диаграмму, показывающую распределение сдавших и не сдавших экзамен
6. Постройте ядерную оценку плотности распределению баллов за экзамен
7. Найдите процентное соотношение учащихся, сдавших экзамен на «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».
8. Какое процентное соотношение юношей и девушек писало данный экзамен?
9. Сколько школ принимало участие в экзамене?

Дополнительные задания

1. Сколько всего заданий с кратким ответом? С развернутым ответом?
2. Пусть задания с кратким ответом будут задания типа В. Соответственно всего по экзамену вопросов класса В: $V_1 \dots V_k$ Посчитайте процент выполненных и невыполненных заданий по каждому вопросу класса В.
3. Аналогично и с типом С (ответы с развернутым ответом)
4. Сделайте анализ по двум школам:
 1. по всем выполненным заданиям типа В
 2. по заданиям типа С больше 50%

3. по среднему баллу юношей и девушек

Контрольные вопросы

1. Какое свойство содержит количество элементов массива по измерениям?
2. Напишите функцию, создающую ndarray из массива [0, 1, 2, 3, 4]
3. Как создать нулевой 4-мерный вектор?
4. Как создать единичную диагональную матрицу 3x4?
5. Сколько операций умножения матриц существует в numpy?
6. Напишите инструкцию, выбирающую каждый третий элемент массива A начиная с 10-го и заканчивая 45-м
7. Напишите инструкцию создания матрицы 4x3 из плоского массива из 12 элементов

ЛАБОРАТОРНАЯ РАБОТА №12 «ВИЗУАЛИЗАЦИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕК MATPLOTLIB И SEABORN»

Цель работы

Научиться строить некоторые виды графиков, используемые в анализе данных, при помощи библиотек matplotlib и seaborn.

Задания для выполнения:

0. Загрузите датасет для предсказания цены квартиры в зависимости от ряда признаков.
1. Постройте круговую диаграмму для признака 'Rooms', иллюстрирующую количество квартир в процентах в зависимости от количества комнат. Сделайте сектор с наибольшим числом квартир выдвинутым.
2. Постройте гистограмму целевой переменной 'Price'. Оцените визуально, по какой цене продаётся наибольшее количество квартир.
3. Постройте диаграммы рассеяния для признаков 'Rooms', 'Square', 'HouseFloor', 'HouseYear' в зависимости от целевой переменной 'Price' в одной области figure(). Оцените визуально, есть ли среди них такие, на которых разброс точек близок к линейной функции.
4. Постройте ядерную оценку плотности целевой переменной 'Price'. Оцените визуально, напоминает ли полученный график нормальное распределение. Постройте двумерную ядерную оценку плотности для целевой переменной 'Price' и признака 'HouseFloor', затем оцените визуально на каких этажах и по какой цене продаётся основная масса квартир.
5. Постройте ящиковую диаграмму признака 'Square'. Оцените визуально имеются ли выбросы, и, если да, то начиная с какого размера площади значение признака можно считать выбросом.
6. При помощи сетки графиков 'PairGrid' визуализируйте попарные

отношения признаков ``Rooms``, ``Square``, ``HouseFloor``, ``HouseYear``, ``Price`` следующим образом: на диагонали - гистограммы, под диагональю - ядерные оценки плотности, над диагональю - диаграммы рассеяния. По результатам визуализации сделайте выводы (аналогичные п.2,3,4).

7. Постройте тепловую карту матрицы корреляции (``df.corr()``) признаков ``Rooms``, ``Square``, ``HouseFloor``, ``HouseYear``, ``Price``. По ней определите, какие признаки являются зависимыми (у таких признаков коэффициент корреляции близок к единице).

Должны быть подписаны названия графиков, названия осей, указаны значения на осях. Оцениваться будет использование количества различных атрибутов при построении графиков и визуальная красота.

Методические указания

Ознакомьтесь с официальной документацией по библиотекам `matplotlib` и `seaborn`.

Дополнительные задания:

1. Постройте график `jointplot` (гибрид `scatterplot` и `histogram`) из библиотеки `seaborn`.
2. Постройте график `violinplot` (гибрид `boxplot` и ядерной оценки плотности) из библиотеки `seaborn`.
3. Поработайте с сеткой подзаголовков `FacetGrid`.

Контрольные вопросы:

1. Построение каких основных видов графиков используется при анализе данных в машинном обучении?
2. В чём разница между библиотеками `matplotlib` и `seaborn`. Преимущества каждой из них.
3. Как задать размер графика в `matplotlib`?
4. Как установить стили в `seaborn`?
5. Для чего используют подграфики `subplots`?

6. Что изображается на ящиковой диаграмме?
7. Как поменять палитру цветов у тепловой карты.

ЛАБОРАТОРНАЯ РАБОТА №13 «ПАРНАЯ ЛИНЕЙНАЯ РЕГРЕССИЯ»

Цель работы

Познакомиться с основными понятиями задачи регрессии и метода градиентного спуска.

Содержание работы

1. Загрузить данные из текстовых файлов.
2. Создать и построить модель парной линейной регрессии с использованием метода градиентного спуска.
3. Оценить на графике качество построенной модели.
4. Построить кривые обучения.

Методические указания

Для полноценной работы с регрессионным анализом данных на потребуются следующие библиотеки языка Python:

```
```py
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```
```

Для начала считаем данные о котировке двух инструментов из csv-файла:

```
```py
x =
pd.read_csv('https://raw.githubusercontent.com/koroteev
mv/ML_course/main/ML1.1_sgd/data/x.csv')
```

```

y
pd.read_csv('https://raw.githubusercontent.com/koroteev
mv/ML_course/main/ML1.1_sgd/data/y.csv')
'''

```

Посмотрим формат файла с данными - состав и названия полей:

```

'''py
x.head()
'''

|index|Unnamed: 0|0| |
|---|---|---|---|
|0|0|1|.462107937044974|
|1|1|1|.1337694423354374|
|2|2|-2|.3015386968802827|
|3|3|1|.74481176421648|
|4|4|0|.0422137467155928|

```

Аналогичная картина наблюдается и в переменной `_y_`. Здесь мы видим две проблемы. Во-первых, при считывании данных из файла `_pandas_` добавил специальный столбец с номером строки. Он нам не нужен. Во-вторых, мы получили объекты типа `_DataFrame_`, а для дальнейшей работы нам нужен объект типа `_Series_`. Эти проблемы можно решить разными способами, но проще всего взять из получившихся датафреймов одну нужную колонку:

```

'''py
x
pd.read_csv('https://raw.githubusercontent.com/koroteev
mv/ML_course/main/ML1.1_sgd/data/x.csv',

```

```

index_col=0) ['0']
y
=
pd.read_csv('https://raw.githubusercontent.com/koroteev
mv/ML_course/main/ML1.1_sgd/data/y.csv', index_col=0) ['0']
...

```

Затем выделим из получившегося объединенного датасета результативный и факторный признак. В нашем случае это будут цены закрытия двух этих инструментов:

```

```py
x.head()
...

...

0      1.462108
1      1.133769
2     -2.301539
3      1.744812
4      0.042214
Name: 0, dtype: float64
...

```

Для визуального представления совместного распределения значений используемых переменных построим график зависимости одной переменной от другой:

```

```py plt.figure()

```

```
plt.scatter(x, y)
plt.show()
...


```

Мы видим, что линейная зависимость прослеживается в данных. Значит, мы можем продолжать и использовать парную линейную регрессию.

Начнем строить класс, который будет реализовывать метод парной линейной регрессии:

```
```py
class Model(object):
    """Модель парной линейной регрессии"""
    def __init__(self):
        self.b0 = 0
        self.b1 = 0
    ...
```

Здесь мы определили конструктор класса, который запоминает в полях экземпляра параметры регрессии. В дальнейшем мы сможем расширять функциональность этого класса сколь угодно много.

Реализуем метод, который принимает значение входной переменной и возвращает теоретическое значение выходной - это прямое действие нашей регрессии - метод предсказания результата по факторам (в случае парной регрессии - по одному фактору):

```
```py
```

```
def predict(self, x):
 return self.b0 + self.b1 * x
```

```
...
```

Теперь зададим функцию ошибки:

```
```py
    def error(self, X, Y):
        return sum((self.predict(X) - Y)**2) / (2 *
len(X) )
    ```
```

В данном случае мы используем простую функцию ошибки - среднеквадратическое отклонение (mean squared error, MSE). Можно использовать и другие функции ошибки. Именно вид функции ошибки будет определять то, какой вид регрессии мы реализуем. Существует много разных вариаций простого алгоритма регрессии. О большинстве распространенных методах регрессии можно почитать в официальной документации `sklearn`.

Теперь реализуем метод градиентного спуска. Он должен принимать массив `X` и массив `Y` и обновлять параметры регрессии в соответствии с формулами градиентного спуска:

```
```py
    def fit(self, X, Y): alpha =
        0.1
        dJ0 = sum(self.predict(X) - Y) / len(X)
        dJ1 = sum((self.predict(X) - Y) * X) / len(X)
        self.b0 -= alpha * dJ0
        self.b1 -= alpha * dJ1
```

```
```
```

Здесь мы используем некоторое произвольное число - *\_alpha\_* - скорость обучения. О выборе конкретного значения *alpha* - скорости обучения мы еще поговорим позднее.

Давайте создадим объект регрессии и проверим начальное значение ошибки:

```
```py
hyp = Model()
print(hyp.predict(0))
print(hyp.predict(100)) J =
hyp.error(x, y)
print("initial error:", J)
```

``` 0
0
initial error: 2937.6446095153397
```
```

Как мы видим, для начала оба параметра регрессии равны нулю. Конечно, такая модель не дает надежных предсказаний, но в этом и состоит метод градиентного спуска: начиная с любого решения мы постепенно его улучшаем и приходим к оптимальному решению.

Теперь изобразим нашу регрессию на графике:

```

```py
X0 = np.linspace(-3, 2, 100) Y0
= hyp.predict(X0) plt.figure()
plt.scatter(x, y) plt.plot(X0,
Y0, 'r') plt.show()
```

```

)

Как и ожидалось, мы видим горизонтальную линию - в нашей примитивной модели пока для любого входного значения, выходное всегда будет 0.

Теперь все готово к запуску градиентного спуска.

```

```py hyp.fit(x,
y)
J = hyp.error(x, y)
print("error after gradient descent:", J)
```

...

error after gradient descent: 2253.745680891855
...

```py
X0 = np.linspace(-3, 2, 100)

```

```

Y0 = hyp.predict(X0)
plt.figure() plt.scatter(x,
y) plt.plot(X0, Y0, 'r')
plt.show()
...

```

)

Как мы видим, численное значение ошибки значительно уменьшилось. Да и линия на графике существенно приблизилось к точкам. Конечно, наша модель еще далека от совершенства. Мы прошли всего лишь одну итерацию градиентного спуска. Модифицируем метод так, чтобы он запускался в цикле определенное количество раз:

```

```py
class Model(object):
 """Модель парной линейной регрессии"""
 def __init__(self):
 self.b0 = 0
 self.b1 = 0
 def predict(self, X):
 return self.b0 + self.b1 * X
 def error(self, X, Y):
 return sum(((self.predict(X) - Y)**2) / (2 * len(X)))
 def fit(self, X, Y, alpha=0.001, accuracy=0.01, max_steps=5000):
 steps, errors = [], []

```

```

step = 0
for _ in range(10000):
 dJ0 = sum(self.predict(X) - Y) /len(X)
 dJ1 = sum((self.predict(X) - Y) * X)

 self.b0 -= alpha * dJ0
 self.b1 -= alpha * dJ1
 new_err = hyp.error(X, Y)
 step += 1
 steps.append(step) er-
 rors.append(new_err)
return steps, errors

```

...

Заодно мы расширили функциональность этого метода так, чтобы он возвращал массив ошибок на каждой итерации спуска и номер итерации. Это будет нам полезно для построения графика обучения.

Запустим наш градиентный спуск:

```
```py
hyp = Model()
steps, errors = hyp.fit(x, y) J =
hyp.error(x, y)
print("error after gradient descent:", J)
```

error after gradient descent: 298.76881676471504
```
```

Как мы видим, теперь ошибка снизилась гораздо больше. Однако, она все еще не достигла нуля. Заметим, что нулевая ошибка не всегда возможна в принципе из-за того, что точки данных не всегда будут располагаться на одной линии. Кроме того, алгоритм градиентного спуска - численный и сходится в пределе, то есть после бесконечного количества повторений. Нужно стремиться не к нулевой, а к минимально возможной ошибке.

Посмотрим, как теперь наша регрессия выглядит на графике:

```
` ``py
X0 = np.linspace(-3, 2, 100) Y0
= hyp.predict(X0) plt.figure()
plt.scatter(x, y) plt.plot(X0,
Y0, 'r') plt.show()
` ``
```


Уже значительно лучше. Линия регрессии довольно похожа на оптимальную. Так ли это на самом деле, глядя на график, сказать сложно, для этого нужно проанализировать, как ошибка регрессии менялась со временем:

```
` ``py plt.figure()
plt.plot(steps, errors, 'g')
plt.show()
```

...

)

На графике наглядно видно, что в начале обучения ошибка падала быстро, но в ходе градиентного спуска она вышла на плато. Учитывая, что мы используем гладкую функцию ошибки второго порядка, это свидетельствует о том, что мы достигли локального оптимума и дальнейшее повторение алгоритма не принесет улучшения модели.

Задания для самостоятельного выполнения

1. Проверьте работу модели с разными начальными значениями параметров. Убедитесь, что модель всегда сходится к оптимуму.

1. Проверьте работу модели с другими значениями скорости обучения. Найдите значение, при котором градиентный спуск расходится.

1. Модифицируйте код модели таким образом, чтобы он корректно работал как с `_Series_`, так и с `_DataFrame_`. При этом, конечно, датафрейм не должен содержать "лишних" столбцов.

1. Модифицируйте алгоритм градиентного спуска таким образом, чтобы он останавливалась, если ошибка перестает снижаться. Для этого сравнивайте, насколько понизилась ошибка на каждом шаге и если это снижение меньше некоторой заранее заданной очень маленькой величины - выходите из цикла. Эту величину можно взять одну миллионную, например.

1. Убедитесь, что алгоритм градиентного спуска завершается всегда -либо если ошибка перестает существенно меняться, либо при достижении максимального количества шагов.

1. Реализуйте механизм адаптивной скорости обучения. Чтобы не задавать вручную параметр `_alpha_` можно отслеживать, увеличивается ли ошибка после первого шага градиентного спуска. В начале задаем скорость обучения близкой к 1. Если ошибка после одного шага увеличивается - скорость обучения надо уменьшить (например, в 2 раза). Если не

увеличивается - скорость обучения можно не трогать. Еще подумайте, как после уменьшения скорости перезапускать градиентный спуск заново.

1. Добавьте к классу модели метод рисования графиков. Сделайте так, чтобы он был независим от масштаба переданных данных.

Контрольные вопросы

1. Сформулируйте, в чем состоит задача регрессии, придумайте несколько примеров.

2. Что такое метод градиентного спуска?

2. Что такое скорость обучения в методе градиентного спуска?

3. Какие функции ошибки используются в регрессионных моделях и почему именно такие?

4. Что показывает конкретное значение ошибки регрессии?

5. Что показывает график обучения и зачем его строить?

Дополнительные задания:

1. Изучите официальную документацию библиотеки [sklearn](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html). Постройте модель регрессии с использованием библиотечных средств на тех же данных. Постройте графики и сравните результаты.

3. Почитайте в документации про встроенный метод оценки качества регрессии score. Используйте его для оценки качества модели. Реализуйте такой же расчет в собственном классе модели.

1. (*) Реализуйте нормирование входных значений. Это нужно делать внутри класса модели. Лучше всего - внутри метода обучения. Необходимо привести значения признаков к шкале [0; 1]. Подумайте, как теперь корректно считать теоретические значения.

ЛАБОРАТОРНАЯ №14: МНОЖЕСТВЕННАЯ ЛИНЕЙНАЯ РЕГРЕССИЯ НА ЧИСТЫХ ДАННЫХ

Цель работы

Познакомиться с основными приемами работы с множественными регрессионными моделями с использованием библиотеки `sklearn`.

Содержание работы

1. Ваша задача - построить предиктивную модель предсказания целевой переменной в зависимости от значения нескольких факторов (признаков).
2. Загрузите датасет из приложения. Вам понадобится два файла -отдельно с признаками (помечен x) и значениями целевой переменной (помечен y). Файлы не содержат заголовков столбцов и строк. Выведите первые строки датасета.
3. Сделайте количественное описание датасета: количество точек данных, признаков, статистику по каждому признаку.
5. Для целевой и каждого из признаков постройте парную регрессионную модель. Визуализируйте линию регрессии и исходные данные на графике. При помощи коэффициентов оцените силу связи.
6. Постройте модель множественной линейной регрессии целевой переменной от всех показателей. Оцените качество регрессии, сделайте вывод.

Методические указания

Отчет по лабораторной работе сдается в виде файла интерактивного ноутбука Jupyter notebook и должен включать весь программный код, реализующий задания лабораторной работы, подробные пояснения, обоснования и выводы там, где это требуется в виде текстовых ячеек.

Для начала работы как всегда выполним стандартный импорт необходимых библиотек:

```

```py
import numpy as np im-
port pandas as pd
import matplotlib.pyplot as plt
```

```

Теперь мы можем приступить к чтению данных. Для этой работы представлено сразу 50 однотипных датасетов. Мы для примера возьмем самый первый (с названием "0"). Читать данные можно многими способами, но мы воспользуемся самым простым чтением по URL:

```

```py
x
=
pd.read_csv("https://raw.githubusercontent.com/korotee
vmv/ML_course/main/ML1.2_regression/data/0_x.csv")
```

```

Теперь нужно вывести датасет (хотя бы первые несколько строк) на экран, чтобы убедиться, что все прочиталось нормально:

```

```py
x.head()
```

```

Если выполнить этот код то при внимательном рассмотрении можно заметить одну ошибку. Первая строка данных была помещена в заголовки строк. Чаще всего в файле с данными первая строка как раз и содержит заголовки, поэтому такое поведение чаще бывает полезным. Но в нашем случае датасет не содержит заголовков. Поэтому при чтении надо это учесть:

```

```py
x =
pd.read_csv("https://raw.githubusercontent.com/korotee
vmv/ML_course/main/ML1.2_regression/data/0_x.csv",
 header=None)

y =
pd.read_csv("https://raw.githubusercontent.com/korotee
vmv/ML_course/main/ML1.2_regression/data/0_y.csv",
 header=None)
```

```

Теперь данные прочтены верно. Но лучше еще раз в этом убедиться, выведя часть датасета на экран.

При работе с данными важно понимать размер данных и то, как они располагаются в структуре, то есть в переменных. Выведем количественные параметры датасета:

```

```py
>>> x.shape, y.shape
((387, 5), (387, 1))
```

```

Здесь мы должны заметить сразу несколько вещей. Во-первых, обе переменные представляют собой двумерные массивы. Точнее они представлены в виде DataFrame. В матрице атрибутов (`_X_`) 387 строк и 5 столбцов. Это значит, что в датасете описаны 387 объектов и у каждого есть 5 признаков. Или, можно сказать, что в `_X_` 5 векторов признаков по 387 значений. Важно то, что по строкам у нас объекты, а по столбцам - атрибуты (характеристики).

В целевой переменной (`_Y_`) у нас один столбец и 387 строчек. Очень важно, чтобы количество строк в обоих переменных совпадало. Фактически, `_Y_` представляет собой вектор-столбец. То есть один единственный атрибут описан для 387 объектов.

Такое представление данных подходит для начала машинного обучения. Все, что нужно совпадает. Теперь мы можем приступать к созданию и обучению модели. Для этого сначала импортируем нужный объект из библиотеки `scikit-learn`:

```
```py
from sklearn.linear_model import LinearRegression
```
```

В данном случае нам нужна модель линейной регрессии, которая является частью пакета `"linear_model"` и называется `LinearRegression`. Про другие модели и пакеты данной библиотеки можно узнать из документации.

Создадим объект модели:

```
```py
binary = LinearRegression()
```
```

Для начала мы построим модель парной регрессии от одного из признаков в датасете. Для этого из матрицы атрибутов нужно вытащить один столбец. Можно сделать это так: `_x[1]_`. Напоминаем, что в `DataFrame` по умолчанию индексация происходит по столбцам. Но тут возникает проблема. Дело в том, что все библиотечные средства предполагают, что `_X_` - это

двумерный массив. А взятие одного столбца возвращает не DataFrame, а Series - одномерный массив. Превратить его в двумерный можно многими разными способами, но в Pandas можно взять несколько столбцов из датафрейма. Такая операция вернет всегда именно датафрейм. А нам нужен датафрейм из одного столбца. Сделать это можно так: `_x[[1]]_`. Итак, возьмем один столбец и обучим парную регрессию его с целевой переменной:

```
```py
binary.fit(x[[1]],
y)
```
```

Данный метод подстроил коэффициенты линейной модели таким образом, чтобы функция модели была как можно ближе к точкам. Сами коэффициенты тоже можно посмотреть. Получаем коэффициенты `p`:

```
```py
binary.coef_
```
```

Метод `predict()` рассчитывает теоретические значения признака:

```
```py
binary.predict(x[[1]])
```
```

Конечно, вручную сравнивать их с реальными (эмпирическими) не нужно. Гораздо проще построить их на графике. Построить точки очень просто:

```
```py
```

```
plt.scatter(x[1], y)
...
```

Для построения линии модели просто использовать plot вот так: `_plt.plot(x[1], y)_` будет уже неправильно. Лучше построить равномерное распределение точек от минимального до максимального по `_X_` и уже по нему рассчитать модельные значения:

```
```py
xx      =      np.linspace(x[1].min(),      x[1].max(),
100).reshape((-1, 1))
plt.plot(xx, binary.predict(xx), c='r')
...

```

Вот что получится в итоге:

! [График
регрессии] (https://github.com/koroteevmv/ML_course/blob/2023/ML1.2_regression/ml12-1.png?raw=true)

Для оценки качества построенной регрессии можно использовать целый ряд показателей. Одним из самых простых и универсальных является коэффициент детерминации (R2-score). Как использовать его читайте в [официальной документации sklearn](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html#sklearn.metrics.r2_score). Методом `score` можно узнать значение метрики эффективности полученной модели:

```
```py reg.score(X,
Y)
```

```
```
```

В нашем случае оценка будет примерно 0.77, что свидетельствует о средней точности модели.

Для построения множественной регрессии выполним те же шаги, но будем использовать полный датасет:

```
```py
multiple = LinearRegression()
multiple.fit(x, y) multiple.score(x, y)
```
```

Теперь точность возросла до 0.99, что говорит о почти идеальной модели. Вообще чем больше информации мы подаем на вход моделям машинного обучения, тем на более сильный результат мы можем рассчитывать.

В модели множественной регрессии уже не удастся построить такой красивый и однозначный график. Для того, чтобы хоть как-то увидеть, как модель соотносится с точками, можно построить график зависимости теоретических значений от эмпирических, то есть как предсказанные значения отличаются от истинных:

```
```py
yy = multiple.predict(x)
plt.scatter(yy, y) plt.plot(yy, yy, c='r')
```
```

Дополнительно на этом графике мы добавили линию $\hat{y} = y$. Это всегда будет прямая, даже если мы будем использовать нелинейные модели. Обратите внимание, что на данном графике по обоим осям Y - то есть целевая переменная:

! [График регрессии] (https://github.com/koroteevmv/ML_course/blob/2023/ML1.2_regression/ml12-2.png?raw=true)

Такой график наряду с диагностическими кривыми можно использовать для диагностики многомерных задач регрессии.

Задания для самостоятельного выполнения

1. Сравните коэффициенты парной и множественной регрессии до и после обучения.

1. Самостоятельно постройте парные регрессии от каждого атрибута с целевой переменной.

2. Постройте сводную таблицу показателей точности всех построенных моделей. Сделайте вывод, какие модели более точные.

3. Используйте другие метрики качества регрессионной модели в дополнение к коэффициенту детерминации.

4. (*) Замерьте время обучения полиномиальных моделей с разной степенью полинома. Сделайте вывод.

Контрольные вопросы

1. Как должны выглядеть данные для модели машинного обучения?

1. В чем физический смысл коэффициентов линейной регрессии?

1. Чем множественная регрессия отличается от парной?

2. Какую функцию ошибки использует модель линейной регрессии?

3. Какие метрики эффективности можно применять для оценки

регрессионных моделей?

4. Какие еще регрессоры есть в библиотеке sklearn?

1. Почему для визуализации модели нельзя использовать предсказанные значения по датасету?

Дополнительные задания

1. Для целевой переменной и каждого из признаков постройте полиномиальную регрессию второго, третьего и десятого порядков. Сделайте вывод о значимости регрессии.

1. (*) Модифицируйте класс модели из предыдущей лабораторной работы таким образом, чтобы он работал в режиме множественной регрессии. Для этого на вход ему в качестве `_X_` должен подаваться датафреймиз произвольного количества строк.

ЛАБОРАТОРНАЯ №15: МЕТОДЫ МЕТРИЧЕСКОЙ КЛАССИФИКАЦИИ

Цель работы

Познакомиться с основными методами метрической классификации, их реализацией и приемами работы в scikit-learn.

Задания для выполнения

1. Загрузите [данные](<https://www.kaggle.com/uciml/pima-indians-diabetes-database>) о диагностике сахарного диабета;
2. Разделите эти данные на тестовую и обучающую выборки;
3. Постройте модель классификации для определения заболевания;
4. Оцените качество модели на тестовой выборке с помощью следующих метрик:
 1. достоверность предсказания (accuracy);
 2. точность (precision);
 3. полнота (recall);
5. Сделайте вывод о применимости модели.

Методические указания

Для начала работы обратимся к набору данных pima-indian-diabetes. Это довольно известный датасет, собравший информацию о медицинских показателях порядка 700 пациентов, обследованных на предмет наличия сахарного диабета. На нем мы потренируемся строить классификационные модели.

Сперва загрузим исходный набор данных. Это можно сделать, как из файла csv (который можно легко найти в публичном доступе), либо используя встроенные функции библиотеки sklearn:

```
` ``py
import pandas as pd
```

```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'in-
             sulin', 'bmi', 'pedigree', 'age',
             'label']

pima = pd.read_csv('diabetes.csv', header=None,
names=col_names)

pima = pima[1:]
```

```

Как и ранее, хорошей идеей перед началом анализа будет познакомиться с составом набора данных визуально. Выведем датасет на экран:

```
```py pima.head()
```
```

При проведении серьезного анализа перед построением модели машинного обучения нужно провести тщательную обработку и очистку набора данных - удаление пропущенных значений, анализ шкал, нормализация, удаление выбросов и аномалий. Используемый нами набор -сам по себе довольно чистый, потому в учебных целях пропустим этот шаг.

Выделим целевую переменную и факторы:

```
```py
Y = pima.label
X = pima.drop(['label'], axis=1)
```
```

Разделим набор данных на обучающую и тестовую выборки:

```

` ``py
from sklearn.model_selection import
train_test_split
x_train, x_test, y_train, y_test =
train_test_split(X, Y,

test_size=0.2,

random_state=True)
` ``

```

В данном случае мы используем удобную встроенную функцию sklearn для разбиения выборки. Мы используем соотношение 80/20 для обучающей и тестовой выборки. Соотношение может колебаться и зависит от множества факторов, но 80/20 - хорошее значение по умолчанию.

Построим модель логистической регрессии:

```

` ``py
from sklearn.linear_model import LogisticRegression cls
= LogisticRegression()
` ``

```

Здесь мы создаем объект классификатора, используя конструктор по умолчанию. Если нам нужно изменить стандартные параметры логистической регрессии (например, задать значение параметра регуляризации), то это можно сделать именно на этом этапе.

Обучим наш классификатор на обучающей выборке:

```
```py
cls.fit(x_train, y_train)
```
```

Для оценки эффективности полученной модели сделаем предсказания по нашему классификатору на тестовой выборке (там, где мы уже знаем правильные ответы, чтобы сравнить, насколько предсказания расходятся с действительностью):

```
```py
y_pred = cls.predict(x_test)
```
```

Оценим эффективность предсказания тестовой выборки при помощи матрицы классификации:

```
```py
from sklearn import metrics met-
rics.confusion_matrix(y_test, y_pred)
```
```

Матрица классификации показывает нам очень полезную информацию: совместное распределение числа объектов предсказанных и реальных классов. Рассматривая эту матрицу мы можем получить важную информацию: сколько объектов мы классифицировали правильно, сколько неверно, к каким классам наша модель тяготеет, какие классы распознаются хорошо, какие - плохо

То же самое в графической форме:

```

```py
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
class_names = [0, 1]
fig,
ax = plt.subplots()
ticks = np.arange(len(class_names))
plt.xticks(ticks, class_names)
plt.yticks(ticks, class_names)
sns.heatmap(pd.DataFrame(
    metrics.confusion_matrix(y_test, y_pred)),
    annot=True)
plt.ylabel('Действительные значения')
plt.xlabel('Предсказанные значения')
```

```

Кроме матрицы классификации весьма полезно использовать численные метрики эффективности классификации. Существует большое количество таких метрик, в данной работе мы сосредоточимся на трех основных.

Метрика достоверности предсказания:

```

```py
metrics.accuracy_score(y_test, y_pred)
```

```

Метрика точности

```
```py
metrics.precision_score(y_test, y_pred)
```
```

Метрика полноты

```
```py
metrics.recall_score(y_test, y_pred)
```
```

### **Контрольные вопросы**

1. Какие выводы мы можем сделать на основании метрик модели, построенной в данной лабораторной работе?
2. Чем логистическая регрессия отличается от линейной?
3. С помощью каких методов происходит оптимизация параметров логистической регрессии?
4. Почему метрики достоверности предсказания недостаточно для оценки эффективности модели?
5. Что такое метрика F1? Какие еще метрики из этого семейства существуют и для чего они служат?
6. Чем метод опорных векторов отличается от логистической регрессии? В чем его достоинства и недостатки?
7. В чем особенности метода ближайших соседей?

### **Дополнительные задания**

1. Используйте метод опорных векторов для построения альтернативной модели предсказания диабета.
2. Сравните метрики эффективности и сделайте вывод о том, какая модель лучше предсказывает заболевание.

3. Постройте модели классификации на основе следующих методов:

1. логистическая регрессия (LogisticRegression);
2. метод опорных векторов с линейным ядром (SVC);
3. метод опорных векторов с гауссовым ядром (SVC);
4. метод k ближайших соседей (KNeighborsClassifier);
5. многослойный перцептрон (MLP);
6. другие методы по желанию;

4. Проанализируйте метрики каждой модели и сделайте выводы об их эффективности и применимости. Сравните эффективность всех этих моделей и выберите лучшую;

5. Для каждой модели из п.3 постройте кривые обучения и диагностируйте недо-/переобучение модели. Попробуйте изменить параметр регуляризации для улучшения результатов модели.

6. Повторите полностью анализ для другой задачи - распознавание вида ириса по параметрам растения (можно использовать метод `sklearn.datasets.load_iris()`).

# ЛАБОРАТОРНАЯ №16: КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ

## Цель работы

Применить методы машинного обучения для решения задач классификации изображений.

## Задания для выполнения

1. Загрузите датасет Olivetti faces;
2. Познакомьтесь с описанием и структурой датасета. Описание можно найти в [документации](<https://scikit-learn.org/stable/datasets/index.html#real-world-datasets>).
3. Выведите информацию о количественных параметрах датасета;
4. Выведите несколько изображений на экран используя инструментальный библиотеки matplotlib;
5. Разделите эти данные на тестовую и обучающую выборки;
6. Постройте модель классификатора метода опорных векторов для идентификации человека по изображению;
7. Оцените качество модели на тестовой выборке с помощью следующих метрик:
  1. достоверность предсказания (accuracy);
  2. точность (precision);
  3. полнота (recall);
8. Постройте кривую обучения - график зависимости тестовой и обучающей эффективности от размера обучающей выборки.
9. Сделайте вывод о применимости модели.

## Методические указания

Датасет Olivetti faces - это один из известных модельных наборов данных для обучения методам классификации изображений. Его можно получить с помощью стандартных средств sklearn:

```py

```
from sklearn.datasets import fetch_olivetti_faces
faces = fetch_olivetti_faces()
print(faces.DESCR)
...
```

При работе с разными данными часто требуется выводить из на экран. Хотя это абсолютно не нужно для работы самих алгоритмов машинного обучения, зачастую проводится, например, анализ ошибок. Для этого нужно уметь выводить изображения стандартными средствами. В библиотеке `matplotlib`, например, можно воспользоваться функцией `imshow`:

```
...py
p.imshow(images[i], cmap=plt.cm.bone)
...
```

Более подробно про эту функцию можно прочитать в [документации](https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.imshow.html).

Методы машинного обучения библиотеки `sklearn` рассчитаны, что на вход будет подаваться плоский массив признаков. Однако, изображение - это двумерный объект. Можно разными способами превратить изображение в набор признаков. В нейронных моделях для этого используются свертки. Но самый простой способ - просто представить изображение как линейную последовательность пикселей.

Если изображение имеет разрешение 64 на 64 пикселя, то всего пикселей получается 4096. Если изображение черно-белое (в градациях серого), то значением признака, соответствующего каждому пикселю можно взять яркость этого пикселя. Если изображение цветное, то каждому пикселю

может соответствовать три разных признака. Причем обычно такие значения нормируются к шкале долей единицы.

Такой набор признаков уже можно использовать как исходные данные для машинного обучения. Далее алгоритм машинного обучения полностью совпадает с работой с численными данными.

Контрольные вопросы

1. Какие выводы мы можем сделать на основании метрик модели, построенной в данной лабораторной работе?
2. Как представляется изображение в методах машинного обучения? От чего зависит размерность вектора признаков?
3. Какие задачи можно решать при помощи анализа изображений? Какие из них самые распространенные?
4. Какие известные датасеты изображений существуют? Для каких задач они применяются?

Дополнительные задания

1. Постройте модели классификации на основе следующих методов:
 1. логистическая регрессия (LogisticRegression);
 2. метод опорных векторов с гауссовым ядром (SVC);
 3. метод опорных векторов с полиномиальным ядром (SVC);
 4. метод k ближайших соседей (KNeighborsClassifier);
 5. многослойный перцептрон (MLP);
 6. другие методы по желанию;
2. Проанализируйте метрики каждой модели и сделайте выводы об их эффективности и применимости. Сравните эффективность всех этих моделей и выберите лучшую;
3. Для каждой модели из п.3 постройте кривые обучения и диагностируйте недо-/переобучение модели. Попробуйте изменить параметр регуляризации для улучшения результатов модели.

4. Сделайте замеры времени обучения для каждой модели. Сделайте вывод о сравнительной производительности моделей.

5. (*) Используйте сверточную нейронную сеть для решения той же задачи. Сравните ее эффективность и производительность с классическими моделями.

ЛАБОРАТОРНАЯ №17: КЛАССИФИКАЦИЯ ТЕКСТОВ

Цель работы

Применить методы машинного обучения для решения задач классификации текстов.

Задания для выполнения

1. Загрузите датасет 20 newsgroups;
2. Познакомьтесь с описанием и структурой датасета. Описание можно найти в [документации](<https://scikit-learn.org/stable/datasets/index.html#real-world-datasets>).
3. Выведите информацию о количественных параметрах датасета;
4. Выведите несколько точек датасета (сами текстовые фрагменты и значение целевой переменной);
5. Разделите эти данные на тестовую и обучающую выборки;
6. Постройте модель наивного байесовского для классификации текстов;
7. Оцените качество модели на тестовой выборке с помощью следующих метрик:
 1. достоверность предсказания (accuracy);
 2. точность (precision);
 3. полнота (recall);
8. Постройте кривую обучения - график зависимости тестовой и обучающей эффективности от размера обучающей выборки.
9. Сделайте вывод о применимости модели.

Методические указания

Датасет 20 newsgroups - это один из известных модельных наборов данных для обучения методам классификации текстов на естественных языках. Его можно получить с помощью стандартных средств sklearn:

```py

```

from sklearn.datasets import fetch_20newsgroups news =
fetch_20newsgroups(subset='all')
...

```

Превращение текста в набор численных признаков - это сама по себе нетривиальная задача. Часто эту задачу называют векторизация текста. От применяемого метода векторизации очень сильно зависит эффективность методов машинного обучения (наверное, даже больше, чем от самих методов классификации).

В библиотеку `sklearn` встроены несколько простых методов векторизации текстов. Они собраны в модуле `sklearn.feature_extraction.text`. Более подробно о них можно почитать в [документации]([https://scikit-learn.org/stable/modules/feature\\_extraction.html#text-feature-extraction](https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction)).

Их применение довольно прямолинейно. Сначала нужно создать векторизатор, а затем преобразовать с помощью него текст в набор векторов:

```

```py
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)
...

```

Обратите внимание, что данный метод преобразования составляет словарь токенов текста. Поэтому если мы применим его на другой выборке у нас получится другой словарь. Чтобы преобразовать другую выборку на основе того же самого словаря нужно применить другой метод:

```

```py
X = vectorizer.transform(['Something completely
new.'])
...

```

Это поведение похоже на обучение модели. Векторизатор обучается на обучающей выборке, а затем используется для преобразования тестовой.

После векторизации исходных данных использование методов машинного обучения ничем не отличается от работы с числовыми данными.

### **Контрольные вопросы**

1. Какие выводы мы можем сделать на основании метрик модели, построенной в данной лабораторной работе?
2. Как представляется текст в методах машинного обучения? Какие основные методы векторизации существуют?
3. Какие задачи существуют в области обработки естественных текстов?
4. (\*) Что такое текстовые вложения (эмбединги)? Как они строятся и зачем применяются?

### **Дополнительные задания**

1. Постройте модели классификации для данной задачи на основе следующих методов:
  1. логистическая регрессия (LogisticRegression);
  2. метод опорных векторов с гауссовым ядром (SVC);
  3. метод опорных векторов с полиномиальным ядром (SVC);
  4. метод k ближайших соседей (KNeighborsClassifier);
  5. многослойный перцептрон (MLP);
  6. другие методы по желанию;
2. Проанализируйте метрики каждой модели и сделайте выводы об их эффективности и применимости. Сравните эффективность всех этих моделей и выберите лучшую;
3. Для каждой модели из п.3 постройте кривые обучения и диагностируйте недо-/переобучение модели. Попробуйте изменить параметр регуляризации для улучшения результатов модели.

4. Сделайте замеры времени обучения для каждой модели. Сделайте вывод о сравнительной производительности моделей.

5. (\*) Используйте глубокую нейронную сеть для решения той же задачи. Сравните ее эффективность и производительность с классическими моделями.

## ЛАБОРАТОРНАЯ РАБОТА №18: «ОЦЕНКА ЭФФЕКТИВНОСТИ МОДЕЛЕЙ БИНАРНОЙ КЛАССИФИКАЦИИ»

### Цель работы

Научиться оценивать эффективность моделей классификации.

### Содержание работы

1. Загрузите данные о вероятности развития сердечного приступа. Описание датасета смотрите здесь:  
<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>.
2. Разделите данные на обучающую и тестовую выборки в соотношении 80% / 20% без перемешивания.
3. Постройте следующие модели классификации:
  1. случайное предсказание (подбрасывание кубика)
  2. логистическая регрессия (LogisticRegression);
  3. метод опорных векторов с линейным ядром (SVC(kernel='linear'));
  4. метод линейного дискриминантного анализа (LinearDiscriminantAnalysis)
  5. метод k ближайших соседей (KNeighborsClassifier с двумя классами).
4. На основе метрики ассигасы сделайте вывод о недообученности-переобученности модели логистической регрессии. Для этого сделайте предсказание на тренировочном датасете и посчитайте ассигасу на нём, затем на тестовом датасете и тоже посчитайте ассигасу. Сравните эти значения, они должны быть примерно одинаковыми. Если точность на обучающей выборке оказывается сильно выше - это означает, что модель переобучилась.
5. Сделайте вывод о применимости каждой модели на основе метрики F1-score (чем выше её значение, тем модель эффективнее).
6. На одном графике постройте ROC-кривые для каждой модели. Визуально определите самую эффективную и наименее пригодную модели. Чем ROC-кривая выше, тем модель более эффективна.

7. Выведите соответствующие значения AUC (площадь под ROC - кривой) и расположите модели в порядке пригодности.

### Методические указания

Загрузим данные в датафрейм `data`.

Целевым значением будет значение `"output"` - случится приступ или нет. Вынесем этот столбец в отдельную переменную `y`, а в переменную `x` запишем все столбцы из `data`, кроме столбца `target`:

```
```py
target = "output" y =
data[target]
x = data.drop(target, axis=1)
```
```

Импортируем нужные библиотеки для вычисления метрик:

```
```py
from sklearn.metrics import accuracy_score,
roc_curve, auc, f1_score

```
```

Вычислим метрику accuracy на тренировочном датасете:

```
```py
y_pred_train = model.predict(x_train) accuracy_score(y_train, y_pred_train)
```
```

Вычислим метрику accuracy на тестовом датасете:

```
```py
y_pred = model.predict(x_test) accuracy_score(y_test,
y_pred)
```

```
```
```

Вычислим метрику F1-score:

```
```py
f1 = f1_score(y_valid, y_pred)
```
```

Алгоритм логистической регрессии позволяет получить не только предсказанную метку класса, но и вероятность принадлежности этому классу. Эти вероятности можно получить с помощью метода `.predict_proba`:

```
```py
y_pred = model.predict(x_test) y_pred_proba =
model.predict_proba((x_test))
```
```

Рассчитываем значения, необходимые для построения ROC-кривой (receiver operating characteristics curve):

```
```py
probs = y_pred_proba[:, 1]
fpr, tpr, treshold = roc_curve(y_test, probs)
```
```

Построить график ROC-кривой можно следующим образом:

```
```py plt.plot(fpr,
tpr)
plt.xlabel('False Positive Rate') plt.ylabel('True
Positive Rate')
```

```
plt.title('ROC-кривая')
plt.show()
...
```

Вычислим значение площади под кривой (AUC-area under curve). Чем лучше классификатор разделяет два класса, тем больше эта площадь и ближе к 1:

```
```py
roc_auc = auc(fpr, tpr)
...

```

### **Задания для самостоятельного выполнения**

1. Изобразите precision-recall кривую.
2. Оцените эффективность моделей при помощи показателя average precision.
  1. Постройте другую классификационную модель и сравните ее с описанными по следующим метрикам: accuracy, precision, recall, ROC-AUC, F1. Сделайте вывод о том, какая модель наиболее удачная для данной задачи.

### **Контрольные вопросы**

1. Что обозначают элементы confusion matrix?
2. По какой формуле рассчитывается метрика accuracy? Что она показывает?
3. Почему метрика accuracy непригодна при дисбалансе классов?
4. По какой формуле рассчитывается метрика precision? Что она показывает?
5. По какой формуле рассчитывается метрика recall? Что она показывает?
6. Какая метрика объединяет в себе precision и recall? По какой формуле она рассчитывается?

7. Какой показатель часто используется для сравнительного анализа нескольких моделей классификации?

8. Зависимость каких величин (при изменении порога) показывает ROC-кривая?

# ЛАБОРАТОРНАЯ РАБОТА №19 ДИАГНОСТИКА И КРИВЫЕ ОБУЧЕНИЯ

## Цель работы

Познакомиться на практике с явлениями недо и переобучения, научиться их выявлять путем диагностики, составлять и применять рекомендации по улучшению эффективности систем машинного обучения

## Задания для выполнения

1. Загрузите датасет `breast_cancer`.
2. Выберите два первых признака. Таким образом мы уменьшим размерность задачи. У нас была задача множественной классификации, а стала - бинарная.
3. Разделите датасет на обучающую и тестовую выборки в пропорции 80-20.
4. Постройте классификатор на опорных векторах с гауссовым ядром с параметром регуляризации  $C=0.01$ . Выведите на экран значение точности на тестовой и обучающей выборке. Сделайте предварительный вывод о степени обученности модели.
5. Постройте графически зависимость тестовой и обучающей точности от размера обучающей выборки. Прокомментируйте получившийся результат. Сделайте вывод о том, недообучается или переобучается модель? В зависимости от этого выберите варианты, которые могут привести к увеличению точности модели (обычно подходит больше одного варианта):
  1. собрать больше объектов наблюдения;
  2. провести обучение на меньшем объеме данных;
  3. исключить часть признаков из модели;
  4. добавить новые признаки в модель;
  5. добавить полиномиальные признаки;
  6. попробовать более сложные модели;
  7. попробовать более простые модели;

8. увеличить регуляризацию модели;
9. уменьшить регуляризацию модели;
6. Проверьте выдвинутую гипотезу и попробуйте улучшить модель в соответствии с выбранными рекомендациями. Проверьте, увеличивается ли тестовая точность. Сделайте вывод.
7. Повторите предыдущий пункт для различных значений  $C$  в диапазоне от 0,0001 до 1 000 000. Обратите внимание на шкалу по вертикальной оси. Для каких значений тестовая точность оптимальна? Какие значения дают недообучение? Какие - переобучение?
8. Постройте график зависимости тестовой и обучающей точности от значения регуляризационного параметра. Сделайте вывод о зависимости степени регуляризации на точность модели.
9. Найдите методом подбора оптимальное значение параметра регуляризации (примерное).
10. Повторите два предыдущих пункта для метода опорных векторов с линейным ядром. Сравните получившиеся результаты. Сделайте вывод о сравнительной эффективности двух этих моделей.

### Методические указания

Для начала работы нам потребуется импортировать необходимые библиотеки:

```
```py
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer from
                                sklearn.model_selection          import
train_test_split
from sklearn.svm import SVC
```
```

В первую очередь загрузим датасет, выберем нужные данные и выделим тестовую выборку для оценки точности модели:

```
```py
breast_cancer = load_breast_cancer()
X = breast_cancer.data y =
breast_cancer.target X_2d =
X[:, :2]
X_train,      X_test,      y_train,      y_test      =
train_test_split(X_2d, y)
```
```

Затем построим первый классификатор. Оценка его точности даст нам приблизительную оценку уровня общей эффективности простых неоптимизированных моделей:

```
```py
clf = SVC(C=0.01)
clf.fit(X_train, y_train) print(clf.score(X_train,
y_train)) print(clf.score(X_test, y_test))
```
```

Сами по себе данные значения мало о чем могут нам сказать. Понятно, что хотелось бы получить как можно более высокие значения. Но скорее всего вы не получите идеальную классификацию, а оценка точности будет лежать в пределах от 0 до 1. Для разных задач разная точность может считаться приемлемой. Для того, чтобы понять, можно ли повысить точность усовершенствованием модели нужно понять, какова степень обученности данной модели. Какую-то информацию может нам дать разница между этими

значениями.

Если тестовая точность близка к обучающей, это скорее всего означает, что модель недообучается и при использовании более сложных моделей (или при снижении регуляризации, что эффективно то же самое) есть надежда улучшить данный показатель.

Если тестовая точность значительно ниже обучающей, это свидетельствует о потенциальном переобучении и потере способности к переобучению. Тогда для совершенствования модели нужно действовать в обратном направлении: использовать более простые модели или увеличивать регуляризацию данной.

Если же тестовая точность выше обучающей, это говорит о наличии случайных ошибок выборки тестовой модели. Такие ошибки неизбежно присутствуют и могут зашумлять результаты диагностики. Поэтому чаще всего для таких работ производится перекрестная проверка, которую мы будем проходить позже.

Для более точного анализа попробуем выяснить, что происходит с точностью при обучении на части обучающей выборки в зависимости от количества используемых точек. Для этого организуем обучение в цикле (обратите внимание на использование функции `linspace`):

```
```py training_scores =  
[] test_scores = []  
numbers = []  
  
for i in np.linspace(20, len(X_train), 20):  
    clf = SVC(C=10, kernel='rbf')  
    X = X_train[:int(i)] y  
    = y_train[:int(i)]  
    clf.fit(X, y)
```

```

        numbers.append(int(i)) train-
            ing_scores.append(clf.score(X_train,
y_train))
        test_scores.append(clf.score(X_test, y_test))

plt.figure(figsize=(10, 5))
plt.plot(numbers, training_scores, label="training
scores")
plt.plot(numbers, test_scores, label="test scores")
plt.legend(loc="best")
...

```

Проанализировав получившийся график мы можем понять более подробную картину. В частности можно видеть, что с ростом объема обучающей выборки точность на ней растет монотонно, а вот точность на тестовой резко возрастает, а затем начинает снижаться.

При анализе таких графиков следует понимать, что на реальных данных всегда будут присутствовать случайные отклонения, что приводит к зашумленности графиков. Это означает, что нужно обращать внимание на общую тенденцию, а мелкие вариации скорее всего не несут никакой полезной информации. Причем, чем меньше выборка в целом, тем больше будет вклад случайных факторов. Поэтому, в общем случае, увеличение объема данных приводит и к более точным моделям и облегчает их анализ.

Для лучшей читаемости результатов рекомендуется зафиксировать минимальное и максимальное значение по вертикальной оси, чтобы все графики, построенные в рамках данной работы были сопоставимы. Например, можно строить все графики в диапазоне y от 0,8 до 1,0.

Контрольные вопросы

1. Как влияет параметр C на степень регуляризации моделей классификации?

2. Что показывает соотношение точности (ошибки) на тестовой и обучающей выборке?
3. Чем характеризуется ситуация недообучения? Как ее распознать на кривых обучения? А переобучение?
4. Каковы рекомендации по улучшению точности модели в случае недообучения? А переобучения?
5. Какова вычислительная сложность подбора гиперпараметров модели?

Дополнительные задания

1. Повторите такой же анализ для полиномиального ядра и степени полинома как гиперпараметра. Прокомментируйте кривые обучения для случаев недо-и переобучения.
2. Исследуйте зависимость степени обученности модели от значения других гиперпараметров различных классификаторов, реализованных в `sklearn`. Постройте кривые обучения в зависимости от значения этих параметров.
3. Повторите данное исследование на полном датасете. Сделайте выводы об оптимальной регуляризации моделей.
4. Повторите данное исследование на другом датасете. Предпочтительно выбирать более объемный набор данных для классификации (несколько тысяч примеров на каждый класс).
5. Напишите функцию, которая автоматизирует процесс подбора гиперпараметров данной модели путем проверки ошибок на тестовой выборке моделей с разными значениями гиперпараметра. Сравните вашу реализацию с функцией `GridSearch`.

ЛАБОРАТОРНАЯ РАБОТА №20 ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ.

Цель работы

Познакомиться с основными приемами предварительной обработки и очистки данных.

Задания для выполнения

1. Загрузите и прочитайте в pandas приложенный датасет. Обратите внимание, что датасет собран «криво», поэтому прежде чем он «правильно» загрузится (рис.1), необходимо его обработать.

2. Сделайте описательную статистику: размер, типы переменных, пустые значения, уникальные имена и т.д.

3. Поработайте с типами переменных. Все числовые значения переведите в формат int.

4. Поработайте со столбцом price. Постройте гистограмму, найдите выбросы (ящичковая диаграмма), удалите аномальные значения, постарайтесь привести к нормальному распределению.

5. Аналогично поработайте и с другими «важными для целевой функции price» параметрами (powerPS, yearOfRegistration, kilometer и т.д).

«Важность» докажите через коэффициент корреляции. Сделайте визуализацию.

6. Постройте ящичковые диаграммы зависимости gearbox, fuelType, vehicleType, notRepairedDamage от price. Сделайте выводы. Можно ли убрать значения с малой частотой? Повлияет ли это на дальнейшее исследование?

7. Найдите пропуски. Удалите столбцы, которые содержат большое количество пропусков. В остальных случаях постарайтесь восстановить пропущенные значения.

8. Сохраните полученный датасет и выведете его.

Методические указания

Реальные данные, используемые для машинного обучения на практике могут происходить из различных источников и содержать в себе ошибки. Эти ошибки - это следствие человеческого фактора, багов в программах обработки данных, неполноты данных в источниках, и множества других причин. Для эффективного использования данных такие ошибки нужно исправить.

Для иллюстрации наиболее распространенных ошибок в данных давайте возьмем фрагмент датасета из задачи про кредитный скоринг:

Пол	
Возраст	
Город	
Доход	
Образование	
Вернул?	

М	
25	
Мск	
70000	

```
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>100000000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
```

<td>A
</td>
<td>28
</td>
<td>СΠ6
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>

</tr>

<tr>
<td> M
</td>
<td>23
</td>
<td>
</td>
<td>40000
</td>
<td>A
</td>
<td>1
</td>

</tr>

<tr>
<td>Ж
</td>
<td>30
</td>
<td> Еκ6
</td>

```
<td>19973
</td>
<td>B
</td>
<td>1
</td>
</tr>
</table>
```

В этом небольшом участке видно сразу несколько несостыковок. Например, сразу же в первом столбце присутствует аномальное значение:

```
<table>
<tr>
  <td>Пол
  </td>
  <td>Возраст
  </td>
  <td>Город
  </td>
  <td>Доход
  </td>
  <td>Образование
  </td>
  <td>Вернул?
  </td>
</tr>
<tr>
  <td> М
  </td>
  <td>25
  </td>
  <td>Мск
  </td>
```

```
<td>70000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>100000000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
```

```

<tr>
  <td>A
</td>
  <td>28
</td>
  <td>CII6
</td>
  <td>80000
</td>
  <td>B
</td>
  <td>0
</td>
</tr>
<tr>
  <td> M
</td>
  <td>23
</td>
  <td>
</td>
  <td>40000
</td>
  <td>A
</td>
  <td>1
</td>
</tr>
<tr>
  <td>Ж
</td>
  <td>30
</td>
  <td> Эк6

```

```

</td>
<td>19973
</td>
<td>В
</td>
<td>1
</td>
</tr>
</table>

```

Далее, во втором столбце 10 - это тоже, скорее всего, неправильное значение. Вряд ли десятилетний ребенок обращался в банк за кредитом:

```

<table>
<tr>
<td>Пол
</td>
<td>Возраст
</td>
<td>Город
</td>
<td>Доход
</td>
<td>Образование
</td>
<td>Вернул?
</td>
</tr>
<tr>
<td> М
</td>
<td>25
</td>
<td>Мск

```

```
</td>
<td>70000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>1000000000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
```

```

</tr>
<tr>
  <td>A
  </td>
  <td>28
  </td>
  <td>СПб
  </td>
  <td>80000
  </td>
  <td>B
  </td>
  <td>0
  </td>
</tr>
<tr>
  <td> M
  </td>
  <td>23
  </td>
  <td>
  </td>
  <td>40000
  </td>
  <td>A
  </td>
  <td>1
  </td>
</tr>
<tr>
  <td>Ж
  </td>
  <td>30
  </td>
  <td>

```

```

        Екб
    </td>
    <td>19973
    </td>
    <td>В
    </td>
    <td>1
    </td>
</tr>
</table>

```

Переходим к третьему столбцу. Здесь все еще очевиднее. В одной из строк значение просто отсутствует. Наверное, его просто забыли внести при заполнении форм, или оно потерялось в процессе преобразования. Для большинства алгоритмов машинного обучения отсутствующие значения недопустимы. Поэтому этот пробел надо будет чем-то заполнить:

```

<table>
  <tr>
    <td>Пол
    </td>
    <td>Возраст
    </td>
    <td>Город
    </td>
    <td>Доход
    </td>
    <td>Образование
    </td>
    <td>Вернул?
    </td>
  </tr>
  <tr>
    <td> М

```

```
</td>
<td>25
</td>
<td>Мск
</td>
<td>70000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>1000000000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
```

```
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>A
</td>
<td>28
</td>
<td>СІІЅ
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
<td> M
</td>
<td>23
</td>
<td>?
</td>
<td>40000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
```

```

</td>
<td>30
</td>
<td> Екб
</td>
<td>19973
</td>
<td>В
</td>
<td>1
</td>
</tr>
</table>

```

В четвертом столбце тоже не все чисто. Присутствует аномально высокое значение. Непонятно даже, откуда оно взялось. Может быть это ошибка, а может быть, какое-то специальное значение (типа MAXINT). Вообще, это может быть просто выбросом, то есть корректным значением, но нехарактерно далеким от среднего. Ведь миллиардеры тоже могут брать кредиты.

```

<table>
<tr>
<td>Пол
</td>
<td>Возраст
</td>
<td>Город
</td>
<td>Доход
</td>
<td>Образование
</td>

```

```
<td>Вернул?
</td>
</tr>
<tr>
<td> M
</td>
<td>25
</td>
<td>Мск
</td>
<td>70000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>1000000000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
```

```
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>A
</td>
<td>28
</td>
<td>СПб
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
<td> M
</td>
<td>23
</td>
<td>
</td>
<td>40000
</td>
<td>A
```

```

        </td>
        <td>1
        </td>
    </tr>
    <tr>
        <td>Ж
        </td>
        <td>30
        </td>
        <td> Екб
        </td>
        <td>19973
        </td>
        <td>В
        </td>
        <td>1
        </td>
    </tr>
</table>

```

Если присмотреться, в этом же столбце есть еще одно подозрительное место. Обратите внимание, что все значения здесь приведены в десятках тысяч. Однако одно конкретное значение приведено с нехарактерной точностью:

```

<table>
  <tr>
    <td>Пол
    </td>
    <td>Возраст
    </td>
    <td>Город
    </td>
    <td>Доход

```

```
</td>
<td>Образование
</td>
<td>Вернул?
</td>
</tr>
<tr>
<td> М
</td>
<td>25
</td>
<td>Мск
</td>
<td>70000
</td>
<td>А
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>100000000
</td>
<td>С
</td>
<td>1
</td>
</tr>
<tr>
```

```
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>A
</td>
<td>28
</td>
<td>СПб
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
<td> M
</td>
<td>23
</td>
<td>
</td>
```

```
 40000 | A | 1 || Ж | 30 | Екб | 19973 | B | 1 |

```

Перейдем к последнему признаку. Здесь вообще без пояснения шкалы непонятно, что эти буквы означают. Самый главный вопрос: можно ли их сравнивать, то есть установлено ли у этого признака отношение порядка. Ведь если это уровень образования (начальный, средний, высший), то их можно представить как числа. А если это направление (техническое, гуманитарное, естественное), то нужно к этому признаку относиться как к категориальному.

```


||
||
||


```

```
<td>Пол
</td>
<td>Возраст
</td>
<td>Город
</td>
<td>Доход
</td>
<td>Образование
</td>
<td>Вернул?
</td>
</tr>
<tr>
  <td> М
  </td>
  <td>25
  </td>
  <td>Мск
  </td>
  <td>70000
  </td>
  <td>А
  </td>
  <td>1
  </td>
</tr>
<tr>
  <td>Ж
  </td>
  <td>31
  </td>
  <td>Мск
  </td>
  <td>100000000
  </td>
```

```
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>A
</td>
<td>28
</td>
<td>СПб
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
```

```

<td> M
</td>
<td>23
</td>
<td>
</td>
<td>40000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>30
</td>
<td> Екб
</td>
<td>19973
</td>
<td>B
</td>
<td>1
</td>
</tr>
</table>

```

Итого в нашем фрагменте несколько сомнительных значений, которые лучше удалить и заполнять более правдоподобными:

```

<table>

```

```
<tr>
  <td>Пол
</td>
  <td>Возраст
</td>
  <td>Город
</td>
  <td>Доход
</td>
  <td>Образование
</td>
  <td>Вернул?
</td>
```

```
</tr>
```

```
<tr>
  <td> М
</td>
  <td>25
</td>
  <td>Мск
</td>
  <td>70000
</td>
  <td>А
</td>
  <td>1
</td>
```

```
</tr>
```

```
<tr>
  <td>Ж
</td>
  <td>31
</td>
  <td>Мск
</td>
```

```
<td>?
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
  <td> M
  </td>
  <td>10
  </td>
  <td>Мск
  </td>
  <td>40000
  </td>
  <td>C
  </td>
  <td>0
  </td>
</tr>
<tr>
  <td>
    ?
  </td>
  <td>28
  </td>
  <td>СПб
  </td>
  <td>80000
  </td>
  <td>B
  </td>
  <td>0
  </td>
```

```

</tr>
<tr>
  <td> M
  </td>
  <td>23
  </td>
  <td>?
  </td>
  <td>40000
  </td>
  <td>A
  </td>
  <td>1
  </td>
</tr>
<tr>
  <td>Ж
  </td>
  <td>30
  </td>
  <td> Екб
  </td>
  <td>19973
  </td>
  <td>В
  </td>
  <td>1
  </td>
</tr>
</table>

```

Отсутствующие значения в числовых признаках чаще всего заменяют на среднее значение по этому признаку. Это дает наименьшее отклонение от

изначального распределения признака:

```
<table>
  <tr>
    <td>Пол
  </td>
    <td>Возраст
  </td>
    <td>Город
  </td>
    <td>Доход
  </td>
    <td>Образование
  </td>
    <td>Вернул?
  </td>
</tr>
<tr>
  <td> М
</td>
  <td>25
</td>
  <td>Мск
</td>
  <td>70000
</td>
  <td> А
</td>
  <td>1
</td>
</tr>
<tr>
  <td>Ж
```

```
</td>
<td>31
</td>
<td>Мск
</td>
<td>49994.6
</td>
<td> C
</td>
<td>1
</td>
</tr>
<tr>
  <td> M
  </td>
  <td>10
  </td>
  <td>Мск
  </td>
  <td>40000
  </td>
  <td> C
  </td>
  <td>0
  </td>
</tr>
<tr>
  <td>
    ?
  </td>
  <td>28
  </td>
  <td>СПб
```

```
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
<td> M
</td>
<td>23
</td>
<td>?
</td>
<td>40000
</td>
<td> A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>30
</td>
<td> ЕкО
</td>
<td>19973
</td>
<td>B
</td>
```

```
<td>1
</td>
</tr>
</table>
```

Хотя иногда используют не среднее арифметическое, а медианное значение:

```
<table>
<tr>
<td>Пол
</td>
<td>Возраст
</td>
<td>Город
</td>
<td>Доход
</td>
<td>Образование
</td>
<td>Вернул?
</td>
</tr>
<tr>
<td> М
</td>
<td>25
</td>
<td>Мск
</td>
<td>70000
</td>
<td>А
</td>
```

```
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>
?
```

<td>28
</td>
<td>CΠ6
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>

| </tr> |
| <tr> |
| <td> M |
| </td> |
| <td>23 |
| </td> |
| <td>? |
| </td> |
| <td>40000 |
| </td> |
| <td>A |
| </td> |
| <td>1 |
| </td> |
| </tr> |
| <tr> |
| <td>Ж |
| </td> |
| <td>30 |
| </td> |
| <td> Эк6 |
| </td> |
| <td>19973 |
| </td> |

```

        <td>B
    </td>
    <td>1
</td>
</tr>
</table>

```

Пропуски в категориальных признаках можно заменять на наиболее популярное (модальное) значение:

```

<table>
<tr>
    <td>Пол
</td>
    <td>Возраст
</td>
    <td>Город
</td>
    <td>Доход
</td>
    <td>Образование
</td>
    <td>Вернул?
</td>
</tr>
<tr>
    <td> М
</td>
    <td>25
</td>
    <td>Мск
</td>
    <td>70000
</td>

```

```
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>?
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
<td>40000
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>
```

M	
	28
	СПб
	80000
	B
	0

	M
	23
	Мск
	40000
	A
	1

	Ж
	30
	Екб

```
<td>19973
</td>
<td>B
</td>
<td>1
</td>
</tr>
</table>
```

Хотя, если пропусков довольно много, то такая замена сильно сместит распределение к центру. Поэтому довольно часто пропуски в категориальных переменных заполняют специальным значением, обозначающим отсутствие значения. Потом, когда мы будем кодировать этот признак, это приведет к созданию еще одной фиктивной переменной. Такой способ сохраняет наибольшее количество информации из исходного набора данных. Он даже может учитывать, если сам пропуск несет какую-то смысловую нагрузку.

```
<table>
<tr>
<td>Пол
</td>
<td>Возраст
</td>
<td>Город
</td>
<td>Доход
</td>
<td>Образование
</td>
<td>Вернул?
</td>
</tr>
<tr>
<td>
```

M	
	25
Мск	
	70000
A	
1	

	Ж
	31
Мск	
C	
1	

	M
	10
Мск	
	40000

```
</td>
<td>C
</td>
<td>0
</td>
</tr>
<tr>
<td>Πp
</td>
<td>28
</td>
<td>CΠ6
</td>
<td>80000
</td>
<td>B
</td>
<td>0
</td>
</tr>
<tr>
<td> M
</td>
<td>23
</td>
<td>Πp
</td>
<td>40000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
```

```

        <td>Ж
    </td>
    <td>30
    </td>
    <td> Екб
    </td>
    <td>19973
    </td>
    <td>В
    </td>
    <td>1
    </td>
</tr>
</table>

```

Еще один способ заполнения пропусков - случайными значениями из распределения данного признака:

```

<table>
<tr>
    <td>Пол
    </td>
    <td>Возраст
    </td>
    <td>Город
    </td>
    <td>Доход
    </td>
    <td>Образование
    </td>
    <td>Вернул?
    </td>
</tr>
<tr>

```

```
<td> M
</td>
<td>25
</td>
<td>Мск
</td>
<td>70000
</td>
<td>A
</td>
<td>1
</td>
</tr>
<tr>
<td>Ж
</td>
<td>31
</td>
<td>Мск
</td>
<td>?
</td>
<td>C
</td>
<td>1
</td>
</tr>
<tr>
<td> M
</td>
<td>10
</td>
<td>Мск
</td>
```

<td>40000
</td>
<td>C
</td>
<td>0
</td>

| </tr> |
| <tr> |
| <td>Ж |
| </td> |
| <td>28 |
| </td> |
| <td>СНБ |
| </td> |
| <td>80000 |
| </td> |
| <td>B |
| </td> |
| <td>0 |
| </td> |
| </tr> |
| <tr> |
| <td> M |
| </td> |
| <td>23 |
| </td> |
| <td>ЕкБ |
| </td> |
| <td>40000 |
| </td> |
| <td>A |
| </td> |
| <td>1 |
| </td> |
| </tr> |

```
<tr>
  <td>Ж
</td>
  <td>30
</td>
  <td> Екб
</td>
  <td>19973
</td>
  <td>В
</td>
  <td>1
</td>
</tr>
</table>
```

Контрольные вопросы

1. Какие основные проблемы в датасете из лабораторной вы обнаружили?
2. Каковы основные методы заполнения пропущенных значений в данных?
3. Какие существуют методы обнаружения аномальных значений в числовых признаках?

Дополнительные задания

1. Постройте модель регрессии для цены автомобиля. Обучите ее на данных, полученных из исходного датасета с применением разных методов заполнения пропущенных значений. Сделайте вывод об эффективности разных методов.

ЛАБОРАТОРНАЯ РАБОТА №21: «ОБРАБОТКА ДАННЫХ ДЛЯ МАШИННОГО ОБУЧЕНИЯ»

Цель работы

Познакомиться с основными приемами дескриптивного анализа данных для подготовки к машинному обучению

Задания для выполнения

1. На сайте [\[https://www.kaggle.com/datasets\]](https://www.kaggle.com/datasets) (<https://www.kaggle.com/datasets>) выбрать набор данных каждому студенту индивидуально. Лучше всего выбирать датасеты в предметной области исходя из личных предпочтений. Не стоит выбирать слишком большие наборы данных (лучше не более 100 Мб).

2. В начале анализа следует привести общую характеристику набора данных: количество наблюдений, признаков, предметная область, содержание признаков, характер целевой переменной, задача обучения (регрессия, классификация, количество классов).

3. Провести в Jupyter средствами pandas, sklearn, seaborn описательный анализ выбранного набора включающего (но не ограниченного) следующие шаги:

1. исследование шкалы измерения каждого существенного признака;
 2. построение эмпирического распределения каждого существенного признака;
 3. анализ количества и распределения отсутствующих значений;
 4. построение совместного распределения каждого признака и целевой переменной;
 5. построение корреляционной матрицы.
4. По каждому пункту анализа необходимо сделать содержательный вывод.
5. Провести подготовительную обработку данных, включающую следующие шаги:

6. удаление или заполнение отсутствующих значений;
7. приведение всех признаков к бинарной либо числовой шкале;
8. удаление несущественных либо избыточных признаков;
9. другие необходимые действия в зависимости от набора данных и задачи (группировка, удаление аномалий, и др.).

6. Количественные характеристики датасета необходимо также привести после его обработки.

Контрольные вопросы

1. Какие существуют методы устранения отсутствующих значений в наборе данных? Исходя из чего нужно применять различные методы?
2. Что показывает коэффициент корреляции? В чем его ограничения?
3. Какие признаки можно считать несущественными?
4. Какие методы преобразования категориальных признаков в численные существуют?
5. Какие статистические распределения чаще всего встречаются в реальных задачах?
6. Что такое совместное и условное распределение?

Дополнительные задания

1. Использовать несколько моделей машинного обучения для решения поставленной задачи.
2. Использовать методику замеры времени обучения каждой использованной модели.
3. Сформировать сводную таблицу результатов обучения, включающую следующие данные:
 1. время обучения модели;
 2. accuracy;
 3. precision;
 4. recall;
 5. f1-score.

ЛАБОРАТОРНАЯ №22: КЛАСТЕРИЗАЦИЯ

Цель работы

Применить методы машинного обучения для решения задач кластеризации

Задания для выполнения

1. Загрузите прилагающийся датасет.
2. Проверьте датасет на наличие пустых значений. Если есть пустые значения, то заменить их на наиболее подходящие.
3. Постройте графическую интерпретацию значений датасета. В качестве x-координаты используйте Annual Income, в качестве y-координаты Spending Score.
4. Определите наилучшее число кластеров с помощью графической зависимости числа кластеров от суммы квадратов расстояний значений датасета до центра кластера.
5. Обучить модель K-средних и предсказать кластер для каждого значения из датасета.
6. Построить графическую интерпретацию из пункта 3, с центрами кластеров.
7. Построить графическую интерпретацию с центрами кластеров, в качестве x-координаты используйте Возраст, в качестве y-координаты Spending Score.
8. Оцените модель с помощью метрик.
9. Сравните эффективность модели кластеризации в зависимости от количества кластеров. Сделайте выводы.

Методические указания

Для начала работы нам потребуется импортировать необходимые библиотеки:

```py

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score,
calinski_harabasz_score
import matplotlib.cm as cm
from sklearn.cluster import KMeans
%matplotlib inline
` ``

```

В первую очередь загрузим датасет, выведем первые несколько строк:

```

` ``py
df = pd.read_csv('.../Mall_Customers.csv')
df.head()
` ``

```

Построим графическую интерпретацию значений датасета. В качестве x-координаты возьмем годовой доход, а в качестве y-координаты оценку трат.

```

` ``py
sns.scatterplot(x='Annual Income (k$)', y =
'Spending Score (1-100)',data=df , hue='Genre')
` ``

```

По графику можно предположить оптимальное количество кластеров для хорошего описания массива.

Проверим наши предположения с помощью перебора количества

кластеров. Для этого построим на графике взаимосвязь суммы квадратов расстояний значений датасета до центра кластера от количества кластеров.

```
```py wcss =
[]
for i in range(1,11):
    k_means = KMeans(n_clusters=i,random_state=42)
    k_means.fit(x)
    wcss.append(k_means.inertia_)
plt.plot(range(1,11),wcss) plt.xticks(range(1,11))
```
```

В качестве количества кластеров возьмем число 5. Создадим модель К-средних и обучим её.

Построим графическую интерпретацию данных с центрами кластеров.

```
```py
plt.scatter(x.Income, x.Score, c=y_kmeans, s=20,
cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 1], centers[:, 2],
c='black', s=200, alpha=0.5);
```
```

## ЛАБОРАТОРНАЯ №23: ПОНИЖЕНИЕ РАЗМЕРНОСТИ

### Цель работы

Применить методы понижения размерности для решения задач машинного обучения.

### Задания для выполнения

1. Загрузите прилагающийся датасет `credit_data`.
2. Проверьте датасет на наличие текстовых атрибутов. Замените текстовые атрибуты на числовые без потери качества данных.
3. Выведите информацию о количественных параметрах датасета;
4. Разделите эти данные на тестовую и обучающую выборки;
5. Обучите модель случайных лесов на обучающей выборке. Проверьте точность предсказаний.
6. Оцените полученную модель с помощью метрик.
7. Понизьте размерность данных с помощью метода главных компонент.
8. Обучите заново модель случайных лесов и оцените ее эффективность с помощью метрик.
9. Постройте график зависимости точности модели от размерности данных.
10. Сделайте вывод о применимости модели.

### Методические указания

Для начала работы нам потребуется импортировать необходимые библиотеки:

```
` ``py
%matplotlib inline
import matplotlib.pyplot as plt im-
port seaborn as sns; sns.set() import
numpy as np
```

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier from
sklearn import metrics
...
```

В первую очередь загрузим датасет, выведем количество пустых значений в каждом столбце:

```
```py
data =
pd.read_csv(r'.../credit_data.csv', delimiter=',')
print(data.isna().sum())
...
```

Удалим столбцы не несущие какой-либо полезной информации, а также вынесем метки строк в отдельный массив target:

```
```py
target = data['Risk']
data = data.drop(['Risk', 'Unnamed: 0', 'Purpose'], axis=1)
...
```

Заменяем текстовые категориальные признаки на числовые с помощью функции map. Пустые значения в столбцах заменим на 0.

```
```py
data['Saving accounts'] = data['Saving
accounts'].map({"little":1, "moderate":2, "quite rich":3
, "rich":4 });
```

```

data['Checking account'] = data['Checking
account'].map({"little":1,"moderate":2,"rich":3 });
target = target.map({"good":1,"bad":0}); data['Saving
accounts'] = data['Saving
accounts'].fillna(0)
data['Checking account'] = data['Checking
account'].fillna(0)
...

```

Заменим полученные категориальные признаки на индикаторы с помощью метода `get_dummies`. Это необходимо поскольку модель случайных лесов плохо работает с категориальными признаками, но неплохо обучается с индикаторами.

```

```py
new_data = pd.get_dummies(data)
new_data.head()
...

```

Нормализуем данные и понизим размерность данных до 2-х атрибутов.

```

```py
from sklearn.cluster import KMeans; from
sklearn.decomposition import PCA;
from sklearn.preprocessing import normalize; y
= KMeans().fit_predict(new_data)
X = normalize(new_data);
x_PCA = PCA(n_components=2).fit_transform(X,2);
print(x_PCA.shape)
...

```

Построим график на основе полученных атрибутов:

```
```py
plt.scatter(x_PCA[:,0], x_PCA[:,1], c=target,
cmap='Spectral')
plt.figure()
```
```

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК ИМЕНИ ПРОФЕССОРА
Н.И. ЧЕРВЯКОВА**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ
по дисциплине
«Биоинформатика»**

для студентов направления подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль) Биология и география

Ставрополь, 2023 г.

СОДЕРЖАНИЕ

| | |
|---|-----|
| <u>ВВЕДЕНИЕ</u> | 184 |
| <u>1 Требования к уровню освоения содержания дисциплины</u> | 186 |
| <u>2 Организационно-методические рекомендации по освоению дисциплины</u>
188 | |
| <u>3 Методические указания по выполнению самостоятельной работы студентов</u> | 188 |
| <u>4 Организация контроля знаний студентов. Формы контроля знаний студентов</u> | 189 |
| <u>5 Рекомендации по работе с литературой и источниками</u> | 191 |
| <u>6 Перечень рекомендуемой литературы</u> | 192 |

ВВЕДЕНИЕ

Дисциплина «Биоинформатика» включена в программу подготовки бакалавров по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль) География и биология и относится к базовой части блока дисциплин.

Дисциплина «Биоинформатика» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин: «Математические методы в биологии и медицине», «Введение в информационные технологии») Дисциплина «Биоинформатика» является базовой для прохождения учебной/производственной практики и написания выпускной квалификационной работы.

Самостоятельная работа – это познавательная учебная деятельность, когда последовательность мышления обучающихся, его умственных и практических операций и действий зависит и определяется самим обучающимся.

В процессе обучения обучающийся должен не только освоить учебную программу, но и приобрести навыки самостоятельной работы. Обучающийся должен уметь планировать и выполнять свою работу.

Целью самостоятельной работы обучающихся, является:

- систематизации и закрепления полученных теоретических знаний и практических умений обучающихся;
- углубления и расширения теоретических знаний;
- формирования умений использовать специальную литературу;
- развития познавательных способностей и активности обучающихся: творческой инициативы, ответственности и организованности при создании собственной научно-исследовательской работы и проектной деятельности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развития исследовательских умений.

Задачи настоящих методических указаний по изучению дисциплин

предусмотренных учебным планом направления подготовки 06.05.01

«Биоинженерия и биоинформатика» включают:

- активацию самостоятельной работы;
- содействие развитию творческого отношения обучающихся к учебе;
- выработку умений и навыков рациональной работы с литературой, Internet- источниками, базами данных;
- обеспечение контроля за ходом самостоятельной работы обучающихся;
- управление познавательной деятельностью обучающихся. Самостоятельная работа обучающихся, способствует развитию самостоятельности, ответственности и

организованности, творческого подхода к решению проблем учебного и профессионального уровня. Этапы самостоятельной работы:

- осознание учебной задачи, которая решается с помощью данной самостоятельной работы;
- ознакомление с инструкцией о её выполнении;
- осуществление процесса выполнения работы;
- самоанализ, самоконтроль;
- проверка работ обучающихся, выделение и разбор типичных преимуществ и ошибок.

Самостоятельная работа является обязательным компонентом учебного процесса для каждого обучающегося и определяется учебным планом. При определении содержания самостоятельной работы следует учитывать уровень самостоятельности и требования к уровню самостоятельности выпускников для того, чтобы за период обучения искомый уровень был достигнут.

1 Требования к уровню освоения содержания дисциплины

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:

| Код, формулировка компетенции | Код, формулировка индикатора | Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов |
|---|---|---|
| ПК-3 Способен формировать развивающую образовательную среду для достижения личностных, предметных и метапредметных результатов обучения средствами преподаваемых учебных предметов | ПК-3.1
Владеет способами интеграции учебных предметов для организации развивающей учебной деятельности (исследовательской, проектной, групповой и др.) | Обучающийся способен эффективно интегрировать содержание предмета при организации исследовательской, проектной и групповой работы, разрабатывая комплексные образовательные задания и оценивая их результативность. |
| | ПК-3.2
Использует образовательный потенциал социокультурной среды региона в преподавании (предмета по профилю) в учебной и во внеурочной деятельности. | Обучающийся способен эффективно внедрять элементы региональной социокультурной среды в образовательный процесс, творчески используя её потенциал при планировании как урочной, так и внеурочной деятельности по профильному предмету. |
| | ПК-3.3
Знает психолого-педагогические условия создания развивающей образовательной среды для достижения личностных и метапредметных результатов обучения | Обучающийся способен создавать и поддерживать развивающую образовательную среду, учитывая психолого-педагогические особенности обучающихся и обеспечивая достижение личностных и метапредметных результатов через применение эффективных педагогических технологий и методов. |
| ПК-11 Способен использовать в профессиональной деятельности теоретико-методические знания дисциплин биологического цикла, | ПК-11.1 разрабатывает элементы содержания образовательных курсов и мероприятий с использованием теоретико-методических знаний дисциплин биологического цикла | Обучающийся способен эффективно разрабатывать содержательные элементы образовательных курсов, интегрируя теоретико-методические знания биологического цикла, создавая со- |

| | | |
|---|--|--|
| практические умения и навыки в области биологии | | временные учебные материалы с учётом актуальных образовательных требований и методических подходов. |
| | ПК-11.2 организует и проводит научные исследования, лабораторные и практические работы в области биологии | Способен эффективно организовывать и проводить научные исследования, лабораторные и практические работы в различных областях биологии, применяя современные методики и подходы для получения достоверных результатов |

Организационно-методические рекомендации по освоению дисциплины

Самостоятельная работа является одним из видов учебной деятельности обучающихся, способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Аудиторная самостоятельная работа по учебной дисциплине на учебных занятиях под непосредственным руководством преподавателя и по его заданию. Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать специфику изучаемой учебной дисциплины, индивидуальные особенности обучающегося.

Контроль самостоятельной работы и оценка ее результатов организуется как единство двух форм:

- самоконтроль и самооценка обучающегося;
- контроль и оценка со стороны преподавателя.

2 Методические указания по выполнению самостоятельной работы студентов

Методические указания включают следующие разделы:

- цель работы;

—задание, которое должно быть выполнено студентом в результате проведения самостоятельной работы;

—варианты заданий;

—основные теоретические положения, необходимые для выполнения задания, они должны быть краткими и содержать ссылки на литературу, в которой эти положения изложены в объеме, достаточном для выполнения самостоятельной работы;

—этапы выполнения задания с указанием конкретных сроков выполнения каждого из этапов и всего задания в целом;

—требования к оформлению графической и текстовой части самостоятельной работы;

—пример выполнения одного из вариантов задания и оформления отчета;

—библиографический список использованных источников.

Для помощи студентам и контроля самостоятельной работы применяются консультации преподавателя.

Консультации являются одной из форм работы кафедры по оказанию педагогически целесообразной помощи студентам в их самостоятельной работе.

Консультации должны носить индивидуальный характер. В случае необходимости разъяснения общих вопросов нескольким студентам проводятся групповые консультации.

Консультации проводятся с целью:

- разъяснения вопросов, возникающих у студентов при самостоятельном изучении учебного материала и выполнении домашних заданий;
- углубления и закрепления знаний по отдельным вопросам и темам дисциплины;
- оказания методической помощи в выборе методов самостоятельной работы.

Посещение консультаций студентов добровольное. Однако преподаватель может организовывать вызовы на собеседование тех студентов, которые в процессе обучения не показывают твердых знаний и не пользуются консультациями по своей инициативе.

На консультации преподаватель объясняет непонятные студентам наиболее сложные вопросы, и указывают литературу, в которой они освещаются.

Консультируя студентов, преподаватель обязан знакомиться с тем, как они изучают рекомендованную литературу, давать указания и советы о методах работы над учебным материалом, способствующие более глубокому и прочному его усвоению

Консультации не должны являться дополнительными занятиями или подменять их. На них запрещается выполнять за студентов или совместно с ними домашние задания. Консультации не должны превращаться в форму натаскивания студентов перед зачетами и экзаменами.

Если в процессе консультации выясняется, что непонимание студентом вопроса связано с недостаточным изучением каких-либо разделов программы, то консультирующий обязан порекомендовать студенту изучить их и проконтролировать степень усвоения этих вопросов.

Консультации не являются проверкой знаний. Знания учебной дисциплины, показанные слушателем на консультации, не должны влиять на оценку знаний на экзаменах и зачетах.

3 Организация контроля знаний студентов. Формы контроля знаний студентов

Контроль и оценка знаний, умений и навыков студентов осуществляется на практических занятиях, консультациях, при приеме экзамена. В ходе контроля знаний преподаватель оценивает понимание студентом содержания дисциплины «Инструментальные средства информационных систем», его способность применять методы управления при решении конкретных задач.

Контроль знаний студентов может осуществляться в следующих формах:

- текущий контроль знаний;
- промежуточный контроль знаний.

Текущий контроль знаний студентов имеет целью:

- дать оценку работы каждого студента по усвоению им учебного материала, выявить недостатки в его подготовке и оказать практическую помощь в их устранении;

Основными формами текущего контроля знаний студентов являются:

- устный контрольный опрос;
- письменный контрольный блиц-опрос;
- защита практического задания;
- промежуточное тестирование;
- проверка конспектов лекций.

Устный контрольный опрос студентов проводится на лекциях (и практических занятиях). По его результатам преподаватель оценивает качество подготовки студента к занятию.

Письменный контрольный блиц-опрос студентов проводится в течение запланированного преподавателем времени на лекциях и практических занятиях путем письменного ответа их на поставленные вопросы, заданные преподавателем. Результаты его проведения отмечаются в журнале. На практических занятиях знания и практические навыки студентов оцениваются по 4-балльной системе. Полученные оценки выставляются в журнале.

При проверке конспектов лекций дается анализ качества их ведения. Отмечаются допущенные ошибки, в рецензии преподавателя оценивается качество конспектирования учебного материала, даются рекомендации по улучшению качества конспектирования лекционного материала.

Промежуточный контроль по дисциплине осуществляется в форме экзамена.

Процедура проведения экзамена осуществляется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования в СКФУ – программам бакалавриата, программам специалитета, программам магистратуры – в СКФУ, Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования – программам аспирантуры, программам ординатуры – в СКФУ.

В экзаменационный билет включаются два вопроса и практическое задание.

Для подготовки по билету отводится 30 минут

При подготовке к ответу студенту предоставляется право пользования Программой дисциплины, средой программирования с языком Python

При проверке практического задания, оцениваются: верность решения, использованные методы решения, логика рассуждений.

4 Рекомендации по работе с литературой и источниками

Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «Инструментальные средства информационных систем», которые прямо или косвенно относятся к обрабатываемой теме.

Изучая литературу и источники, студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные на ваш взгляд мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, студенту следует выписать их в виде цитат с полным указанием библиографических источников.

Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо выписывать все выходные данные по используемой литературе и источникам. Важным этапом при работе с рекомендуемой литературой и источниками является изучение законодательных и нормативных актов федерального, регионального, местного и ведомственного уровней. При изучении Указов Президента РФ, Законов и Кодексов РФ, постановлений, положений, рекомендаций и т.д., студент должен выяснить все изменения и дополнения, которые могли быть внесены после их выхода в свет.

Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «Инструментальные средства информационных систем».

Работа с учебно-иллюстрационными материалами имеет следующие этапы.

1. Изучение теоретических основ учебного материала в аудитории изложение преподавателем изучаемого материала студентам с объяснением по опорному конспекту;

2. Самостоятельная работа: индивидуальная работа студентов по опорному конспекту; фронтальное закрепление по блокам опорного конспекта.

3. Первое повторение – воспроизведение содержания заданной темы опорного конспекта по памяти.

4. Устное проговаривание материала опорного конспекта –необходимый этап внешнеречевой деятельности при усвоении учебного материала.

5. Второе повторение – взаимопрос и взаимопомощь студентов друг другу.

Применение учебно-иллюстрационных материалов позволяет обобщить сложный по содержанию материал, активизировать мыслительную деятельность студентов.

Необходимо помнить, что главное для студента в самостоятельной работе с рекомендуемой литературой и источниками - это формирование своего индивидуального стиля, который может стать основой в будущей профессиональной деятельности.

5 Перечень рекомендуемой литературы а)

основная литература:

1. Жукова, И. В. Биофизические основы сложных систем : учебное пособие : [16+] / И. В. Жукова, И. А. Валеев ; Казанский национальный исследовательский технологический университет. – Казань : Казанский национальный исследовательский технологический университет (КНИТУ), 2020. – 100 с. : ил., табл. – Режим доступа: по подписке. –URL: <https://biblioclub.ru/index.php?page=book&id=700147> (дата обращения: 20.04.2023). – Библиогр. в кн. – ISBN 978-5-7882-2831-0. –Текст : электронный.

б) дополнительная литература

1. Frazer J. et al. Large-scale clinical interpretation of genetic variants using evolutionary data and deep learning //bioRxiv. – 2020. – С. 2020.12. 21.423785.
2. Жорина, Л. В. Основы взаимодействия физических полей с биообъектами : использование излучений в биологии и медицине : учебник / Л. В. Жорина, Г. Н. Змиевской ; под ред. С. Щукина ; Московский государственный технический университет им. Н.Э. Баумана. – Москва : МГТУ им. Н.Э. Баумана, 2014. – 376 с. : ил., табл. –(Биомедицинская инженерия в техническом университете). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=575926> (дата обращения: 20.04.2023). – Библиогр.: с. 369. – ISBN 5-7038-3937-9. – Текст : электронный.

3. Практикум по биофизике : в 2 частях : [16+] / под ред. А. Б. Рубина, Г. В. Максимова, С. М. Ременникова. – 2-е изд., электрон. – Москва : Лаборатория знаний, 2020. – Часть 2. – 512 с. : ил.,табл., схем. – (Учебник для высшей школы). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=595197> (дата обращения: 02.05.2023). – ISBN 978-5-00101-773-8. – ISBN 978-5-00101-775-2 (Ч. 2).
– Текст : электронный.

2.

в) методическая литература

1. Методические рекомендации к проведению практических работ по дисциплине «Биоинформатика»

г) программное обеспечение и Интернет ресурсы

1. <http://www.cellbiol.ru/book/bioinformatika> Справочник по биоинформатике
2. <http://sbio.info/> Портал «Вся биология»
3. Maple, Microsoft Office Standard 2013 (договор №01-эа/13 от 25.02.2013, лицензия №61541869)
4. Microsoft Windows 10 Prof (договор № 27-эа/16 от 02.08.2016, лицензия № 69366049)

д) Материально-техническое обеспечение дисциплины

1. ПЭВМ с микропроцессором не ниже Pentium 4 объем свободной памяти в ПЗУ не менее 3 Гб, объем ОЗУ не менее 2 Гб. Мультимедийный проектор.