

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
Московский государственный университет имени М.В. Ломоносова
Факультет биоинженерии и биоинформатики

УТВЕРЖДАЮ

Декан
факультета биоинженерии
и биоинформатики,
академик

_____/В.П. Скулачев /

« ____ » _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины:

Практическая информатика

Уровень высшего образования:
специалитет

Направление подготовки (специальность):

06.05.01 Биоинженерия и биоинформатика

Форма обучения:

очная

Рабочая программа рассмотрена и одобрена

Ученым советом факультета

(протокол № _____, _____)

Москва 20__

Рабочая программа дисциплины разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по специальности 06.05.01 «Биоинженерия и биоинформатика» (программы специалитета) в редакции приказа МГУ от 30 декабря 2016 г.

Год (годы) приема на обучение – 2016, 2017, 2018, 2019.

© Факультет биоинженерии и биоинформатики МГУ имени М.В. Ломоносова

Программа не может быть использована другими подразделениями университета и другими вузами без разрешения факультета.

Цель и задачи дисциплины

Цель курса - научить студентов базовым навыкам владения компьютером на примере анализа биологических последовательностей.

Задачи курса:

- овладение основами: внутренне устройство компьютера, операционная система, файловая система, оболочка FAR manager
- знакомство с геномом как с файлом
- основы программирования на Python
- основы языка HTML, создание учебного веб сайта
- использование программы Excel и аналогичных программ для анализа данных

1. Место дисциплины в структуре ОПОП ВО – вариативная часть, гуманитарный, социальный и экономический цикл, курс I – семестр 1.

2. Входные требования для освоения дисциплины, предварительные условия (если есть): желательно (но не обязательно) освоение информатики в рамках школьной программы

3. Планируемые результаты обучения по дисциплине:

Знать:

- основы операционных систем Windows и Linux

Уметь

- использовать базовые утилиты Linux
- написать простую программу на языке python
- использовать функции программы Excel для обработки таблиц данных

Владеть:

Excel

Иметь опыт

- анализа молекулярно - биологических данных

4. Формат обучения: практические занятия

5. Объем дисциплины составляет 2 з.е., в том числе 36 академических часов, отведенных на контактную работу обучающихся с преподавателем, 36 академических часов на самостоятельную работу обучающихся.

6. Краткое содержание дисциплины (аннотация):

Раздел 1. Файловая система. FAR. Геном как файл.

Тема 1. Что такое биоинформатика?

Тема 2. Устройство компьютера. Far Manager. WinSCP

Тема 3. Вирусные геномы как файлы. Знакомство с форматом GenBank.

Тема 4. Прокариотические геномы. Геномные браузеры.

Раздел 2. Основы программирования на Python.

Тема 1. Введение в программирование на Python.

Тема 2. Условия, списки, циклы, чтение файла.

Тема 3. Классы.

Тема 4. Чтение файла в формате FASTA.

Тема 5. Пример скрипта, создающего и использующего класс.

Тема 6. Словари.

Раздел 3. Основы языка HTML.

Тема 1. Создание персональной веб-странички для представления результатов по предмету.

Раздел 4. Электронные таблицы.

Тема 1. Программа MS-Excel.

Наименование и краткое содержание разделов и тем дисциплины, Форма промежуточной аттестации по дисциплине	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы			Самостоятельная работа обучающегося, часы (виды самостоятельной работы – эссе, реферат, контрольная работа и пр. – указываются при необходимости)
		Занятия лекционного типа	Занятия семинарского типа	Всего	
Раздел 1. <i>Файловая система. FAR. Генетика как файл.</i>	16	8	0	8	8
Раздел 2. <i>Основы программирования на Python.</i>	20	12	0	12	8
Раздел 3. <i>Основы языка HTML.</i>	16	8	0	8	8
Раздел 4. <i>Электронные таблицы.</i>	16	8	0	8	8
Промежуточная аттестация - зачет	4				4 (количество часов, отведенных на промежуточную аттестацию)
Итого	72	36		36	36

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

1. Входных параметров нет. Однако известно имя файла в указанной выше директории /cw1 на диске P: файл называется codons.txt. В нем таблица с колонками: кодон; однобуквенное обозначение аминокислотного остатка; трехбуквенное обозначение аминокислотного остатка. В строке с названиями колонок первый символ – “#”; эту строку следует игнорировать. Программа должна создать в своей директории credits файл aas-codons-count.txt с таблицей; в таблице две колонки: one_letter_aa_code и codons_count. Поля должны быть разделены символом табулятора ‘\t’. Codons_count – число кодонов, кодирующих данную аминокислоту. Строки выходного файла должны быть упорядочены по алфавиту однобуквенных кодов аминокислотных остатков.

2. Программа получает один аргумент командной строки – натуральное число n более 10. Программа возвращает в выходной поток сообщение

«Сумма простых чисел $\leq n$ равна »

Здесь и далее на месте - ответ. Проверяться будет, например, для $n = 1000$.

3. Программа получает один аргумент командной строки – имя файла в fasta формате, содержащего несколько нуклеотидных последовательностей. Не забудьте, что длинные последовательности записываются в несколько строк. Программа возвращает новый файл с именем seq-lengths.txt такого формата:

```
#Sequence_name \t sequence_length
seq1 \t 125
seq2 \t 1503
```

Совет. Используйте файл test.fasta для отладки программы. Для окончательной проверки используйте файл mosses-mito.fasta. Выходной файл для этого fasta файла и оставьте в директории credits.

4. Программа получает один аргумент командной строки – имя файла в формате .gbk – одного из файлов из директории /P/y18/term1/block3/gbk. Программа выдает количество белков, закодированных на обратной (complement) цепи. Белок находится по слову CDS в колонках 6-8, до него пять пробелов и после него 13 пробелов. В этой же строке есть слово complement – если белок закодирован на комплементарной цепи, а не на прямой.

5. Программа получает два аргумента командной строки – код файла в PDB-формате и десятичное число. Программа подсчитывает число количество атомов, в которых X-координата меньше второго аргумента и выдает сообщение вида: "В pdb файле имеется атомов с координатой X меньше". Pdb код берете из первого аргумента командной строки, порог координаты X – из второго аргумента с точностью до одного знака после запятой, число атомов подсчитываете.

Координаты атомов указаны в тех строках, которые в первых шести колонках содержат либо слово "АТОМ ", либо слово "НЕАТОМ". Координата X атома указана в колонках 31-38. См. подсказки к практикуму 10, в которых описано как прочитать pdb файл из pdb, зная pdb код.

6. Программа работает с файлом /P/y18/term1/block3/pdb/1bfp.pdb. Можно этот файл скачать в свою директорию credits. Вместо этого файла можно взять любой другой из указанной директории. Программа получает два аргумента командной строки – серийный номер атома – натуральное число; порог расстояния – десятичное число. Серийный номер атома содержится в строке АТОМ или НЕАТОМ в позициях 7-11. Пример вызова программы:

```
python3.6 cw1_5.py 137 5.0
```

У всех атомов серийный номер уникален, он идет последовательно от 1 до последнего атома. Программа выдает в выходной поток количество атомов в структуре, находящихся на расстоянии менее указанного во втором аргументе командной строки, от указанного атома. Координата x атома указана в позициях 31-38, y - в позициях 39-46, z - в позициях 47-54. Все координаты измеряются в ангстремах, пространство евклидово.

Программа должна выдать сообщение вроде такого

" angstrom neighborhood of the atom no. of pdb file 1bfp.pdb contains atoms".

7. Программа получает три аргумента командной строки – имя файла в формате .fasta из директории /P/y18/term1/block3.fasta/mosses-mito-split, номер нуклеотида в последовательности, нечетное натуральное число – размер "окна". Все файлы в этой директории содержат одну нуклеотидную последовательность. Последовательность занимает больше одной строчки. Программа возвращает фрагмент последовательности, соответствующий размеру окна с центром

в заданном нуклеотиде. При ошибке входных данных – номер нуклеотида больше, чем длина последовательности, "окно" выходит за границы последовательности, выдается соответствующее сообщение (например, “Error: nucleotide number is more than sequence length”; “Error: window goes beyond sequences boundaries”).

<http://kodomofbb.msu.ru/wiki/2019/1>,

<http://kodomofbb.msu.ru/wiki/2018/1>,

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

Варьируют от года к году в связи с ежегодными нововведениями в программе

Шкала и критерии оценивания результатов обучения по дисциплине.

Результаты обучения	«Неудовлетворительно»	«Удовлетворительно»	«Хорошо»	«Отлично»
Знания: - основ операционных систем Windows и Linux	Знания отсутствуют	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения: - использовать базовые утилиты Linux - написать простую программу на языке python - использовать функции программы Excel для обработки таблиц данных	Умения отсутствуют	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
Владения: Excel	Навыки владения отсутствуют	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

8. Ресурсное обеспечение:

- Перечень основной и дополнительной литературы
- Перечень лицензионного программного обеспечения (при необходимости)
- Перечень профессиональных баз данных и информационных справочных систем
- Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости)
- Описание материально-технического обеспечения.