1. Здравствуйте Меня зовут Артемьев Святослав, я хочу представить вашему вниманию проект на тему “Угадай модель эволюции”. Научным руководителем этой работы является Биба Дмитрий.
2. Модель эволюции представляет собой таблицу размером 4x4, где каждая ячейка содержит число, пропорциональное интенсивности замен каждого нуклеотида на другой. Например, если все нуклеотиды заменяются с одинаковой частотой, то все числа в таблице будут одинаковыми. Если же гуанин чаще всего заменяется на аденин, то число в соответствующей ячейке будет выше, чем в остальных. Знание модели эволюции может быть полезным для определения риска возникновения мутаций у пациента в случае рака определенного типа. Задачи: 1) Написать программу, которая будет изменять последовательность ДНК согласно заданной модели эволюции. 2)Написать программу, которая будет угадывать модель эволюции на основании следующей информации: последовательность ДНК предка, последовательность ДНК потомка, и время, на протяжении которого эта последовательность эволюционировала. 3)Оценить работу программы-предиктора.
3. В нашей работе основополагающую роль имеют цепи Маркова. Потому, чтобы все могли понять, о чём именно моя работа, остановимся на понятии “цепи Маркова” подробнее. Давайте представим ситуацию, что у нас есть город, в котором, если сегодня солнечно, то завтра будет солнечно с вероятность 0.5, будет облачно с вероятность 0.3 и дождь с вероятность 0.2. Если сегодня облачно, то с завтра будет солнечно с вероятность 0.3, однако на следующий день точно не может быть также облачно, а вероятность того, что завтра будет дождь — 0.7. Если сегодня идет дождь, то завтра будет солнечно с вероятность 0.3, будет облачно с вероятность 0.3 и дождь с вероятность 0.4. Соответственно всё это можно представить в формате графа, а также можно представить в виде матрицы P. В ней строчки — начальное состояние, столбцы — конечное. И здесь важно сделать акцент, что нам неважно, какая погода была позавчера. Важно только то, какое состояние сейчас. Именно из-за этого и идёт идея Марковских “цепей” — для каждого из звеньев цепи характерно, что последующее звено зависит только от существования звена в настоящий момент. Потому, если порвать цепочку в каком-то месте, большинство звеньев никак не изменятся. Продолжим рассматривать наш пример с погодой. Какие вероятности переходов мы ожидаем увидеть через 2 дня? 3 дня? Для того, чтобы это вычислить нам достаточно возвести эту матрицу в количество шагов. То есть, например, если мы возведем матрицу P во 2-ю степень, то вероятность того, что во 2-й день было солнце и в 3-й день будет солнце равна: 0.5 · 0.5 + 0.3 · 0.3 + 0.2 · 0.3 = 0.25 + 0.09 + 0.06 = 0.4 по правилу умножения матриц. Рассмотрим ещё один момент: Представим, что по какой-то причине мы знаем, что завтра будет со 100% вероятностью солнечно. Тогда как будет выглядеть матрица вероятностей? В таком случае все строки, кроме солнечной, будут забиты 0. А если, скажем, мы знаем, что вероятность того, завтра солнечный день — 0.8, облачный — 0.1 и дождливый — 0.1. Как тогда должна выглядеть матрица вероятностей? Для того, чтобы это понять мы просто должны перемножить эти матрицы между собой и получить результат.Соответственно это мы будем называть стартовым или начальным распределением.
4. В нашем проекте используются цепи Маркова с дополнением, касательно непрерывного времени. И, для этого дополнения, характерны следующие свойства: 1) Наличие Q-матрицы. Рассмотрим пример. Пусть у нас есть матрица, отображающая интенсивности замен нуклеотидов. То есть аденин заменяется на тимин с интенсивностью 1 замена делёная на какое-то условное время, т.е. нуклеотид/условное время. Такую матрицу мы в дальнейшем и будем звать матрицу интенсивностей или Q-матрицу. По сути своей, все модели изначально предполагают именно Q-матрицу. 2) Так как нам привычнее и удобнее работать с матрицами вероятностей, где сумма в каждой строке равна 1 (так как нуклеотид должен куда-то да перейти или остаться самим собой), то для того, чтобы перейти от матрицы интенсивностей к матрице вероятностей, нужно использовать следующую формулу: P(t) = e^Qt. Соответственно итоговую матрицу вероятностей можно также представить в виде графа, что показано на картинке слева.
5. Итак, теперь рассмотрим сами модели эволюции (по сути просто вариации Q матриц). Первая и самая простейшая — модель Джукса-Кантора. Допущения модели состоят в том, что: Интенсивности замен всех нуклеотидов равны; Равновесные частоты всех нуклеотидов равны.
6. Следующая модель — трехпараметрическая модель Кимуры (K81). Трехпараметрическая модель. Допущения модели:

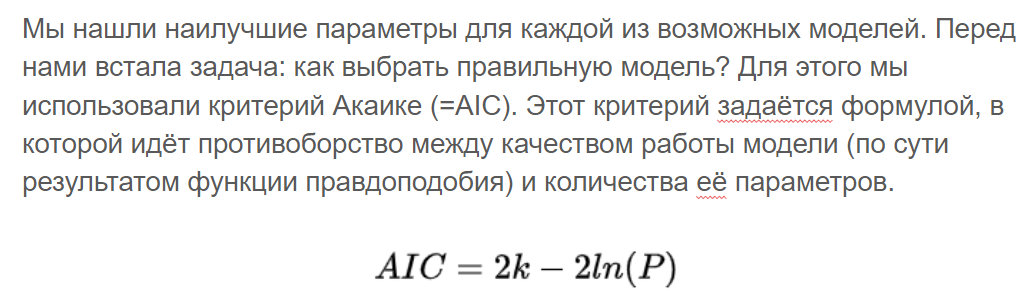
Разные показатели интенсивностей переходов для транзиций и 2-х типов трансверсий.

* 1. Переменная альфа отвечает за показатель интенсивностей для транзиций, то есть переходов A<—>G, C<—>T.
  2. Переменная бета отвечает за показатель интенсивностей для 1-го типа трансверсий, то есть переходов A<—>C, G<—>T
  3. Переменная лямбда отвечает за показатель интенсивностей для 2-го типа трансверсий, то есть переходов A<—>T, G<—>C

Равновесные частоты всех нуклеотидов равны

1. Третья модель — модель Фельзенштейна. По своей сути это расширение модель JC69, однако в этой модели допускается, что равновесные частоты нуклеотидов могут быть неравны.
2. Последняя и самая сложная модель.
   1. Интенсивности замен для каждой отдельно взятой пары нуклеотидов разные, однако вероятности перехода нуклеотидов внутри пары равны.
      1. Переменная альфа отвечает за вероятность переходов A<—>G
      2. Переменная бета отвечает за вероятность переходов A<—>C
      3. Переменная лямбда отвечает за вероятность переходов A<—>T
      4. Переменная дельта отвечает за вероятность переходов C<—>G
      5. Переменная эпсилон отвечает за вероятность переходов C<—>T
      6. Переменная эта отвечает за вероятность переходов G<—>T
   2. Равновесные частоты всех нуклеотидов равны
3. Теперь обобщим все наши сведения и разберёмся в итоговой схеме работы проекта. На данном слайде представлена идея работы программы-генератора. Ей на вход нужно подать модель эволюции и параметры к ней, последовательность гена, к которому мы хотим применить эволюцию, а также указать время эволюции. В промежуточных шагах программы стоит следующее: сгененировать Q-матрицу, исходя из параметров модели, а затем сгенерировать P-матрицу, используя Q-матрицу и заданное время. На выходе мы получаем сгенерированную последовательность нуклеотидов, которую можно сохранить в fasta-файл.
4. Рассмотрим схему работы программы-предиктора. Ей на вход требуется последовательности генов предка и потомка, а также время, за которое происходила эволюция. Промежуточными шагами программы является вычисление эмпирической матрицы вероятностей P(t), после, с учётом времени, вычисление матрицы Q. После, понимая эмпирическую Q-матрицу мы хотим выбрать наиболее подходящую под наш случай модель. А для того, чтобы у нас было из чего выбирать, для начала, нам нужно подобрать подхощящие параметры ДЛЯ КАЖДОЙ ИЗ 4Х МОДЕЛЕЙ. Чтобы это сделать, мы будем использовать градиентный спуск. После нам останется выбрать подходящую модель, мы будем это делать с помощью информационного критерия Акаике.
5. Градиентный спуск — численный метод нахождения локального минимума или максимума функции с помощью движения вдоль градиента, один из основных численных методов современной оптимизации. Пример реализации алгоритма для нахождения точки минимума функции y = x^2.
6. Выберем стартовую точку случайным образом. Пусть position = 10.
7. Найдем производную в окрестностях этой точки. (Мы взяли dx = 0.0001).
8. Пересчитаем position, с учетом полученной производной: Position = Position - Z \* f’(position) , Z — калибровочный коэффициент. (у нас Z = 0.1)
9. Повторяя много раз, position будет стремиться к точке минимума.

Соответственно таким образом мы можем искать нули функции.

1. \_Для того, чтобы подбирать параметры за разумное время нам и понадобится градиентный спуск, ведь пользуясь только перебором — количество необходимых вычислений было бы очень большим. Чтобы использовать градиентный спуск в нашем проекте, мы искали нули следующего выражения: эмпирическая матрица минус теоретическая в квадрате, где теоретическая матрица считается по заданным параметрам. Чтобы расширить градиентный спуск на несколько переменных достаточно ввести несколько стартовых позиций для каждой переменной, и считать производную относительно каждой переменной, после чего пересчитывать эти самые стартовые переменные.
2. \_

В уравнении k — количество параметров модели, P — результат функции правдоподобия. В нашем случае, функция правдоподобия основывается на биномиальном распределении. Обобщая, основная идея этого критерия в противоборстве количества параметров и качества модели.

1. Теперь можно приступить к демонстрации работы наших программ.
2. В заключении хочется сказать, что все поставленные задачи были выполнены. Помимо этого исследование качества программы-предиктора показало, что:

* При time в промежутке 0.05 до 3.5. Тогда процент сбоев составляет в среднем 0,047%, а средний процент угадываний — 91,125%
* Наблюдается закономерность, чем меньше время — тем чаще программа-предиктор будет выбирать простые модели.

Спасибо за внимание, на этом я заканчиваю свой доклад, готов ответить на ваши вопросы.