Цель работы: целью данной работы является разработка генетического алгоритма множественного выравнивания биологических последовательностей

Описание алгоритма

Вход: множество последовательностей $S_1,\ldots,S_n$, которые требуется выровнять, где n – количество входных последовательностей.

Выход: множественное выравнивание входных последовательностей

Общая схема генетического алгоритма

1. Создание множества случайных позиционно весовых матриц, являющихся «организмами» для генетического алгоритма – начальная популяция.
2. Модификация всех организмов
3. Расчет значений целевой функции для каждого организма
4. В случае если популяция стабилизировалась, либо было достигнуто предельное число итераций, переход к шагу 9, иначе переход к шагу 5
5. Проведение селекции среди особей.
6. Скрещивание и образование нового поколения потомков и мутации среди них
7. Модификация всех организмов
8. Расчет значений целевой функции для поколения потомков, возврат к шагу 4
9. Выбор матрицы с наибольшим значением числовой функции
10. Расчёт статистической значимости матрицы
11. Построение итогового выравнивания, **конец алгоритма**

Организм

Позиционно весовой матрицей называется матрица, которая может быть построена по множественному выравниванию по следующему принципу. Строкам данной матрицы будут соответствовать значения аминокислот, а столбцам позиции в исходных последовательностях. Сначала строится позиционная матрица частот встречаемости определенной аминокислоты на выбранной позиции в заданном множественном выравнивании – $M^{20\times L}$. Из матрицы частот можно получить вероятностную матрицу – $P^{20\times L}$, делением каждого элемента столбца матрицы P, на сумму элементов столбца M.

$$\text{Для } i = 1,\ldots,20$$

$$P_{i,j} = \frac{M_{i,j}}{X_i}$$

$$\text{где } X_i = \sum_{j=1}^{20} M_{i,j}$$

Далее для вычисления по заданной вероятностной матрице – $P^{20\times L}$, соответствующей ей позиционно весовой матрицы – $W_{k,j}$ вычисляется логарифмическое отношение правдоподобия. Пусть P - полученная на предыдущем шаге вероятностная матрица, в таком случае расчет значений соответствующей ей позиционно весовой матрицы выглядит как:

$$\text{Для } i = 1,\ldots,20;\ j = 1,\ldots,L;\ k = 1,\ldots,L$$

$$W_{k,j} = ln\frac{P_{k,j}}{P_k}$$

$$\text{где } P_k = \frac{y_i}{\sum_{l=1}^{20} y_l}$$

Таким образом, можно построить по множественному выравниванию последовательностей соответствующую этому выравниванию позиционно весовую матрицу – $W^{20\times L}$. Здесь и далее в работе под организмом будет подразумеваться позиционно весовая матрица размерностью 20×L, где L – длина последовательностей $S_1,\ldots,S_n$.

Инициализация

Все позиционное весовые матрицы организмов в данной работе имеют размерность 20×L, где L – длина последовательностей $S_1, \dots, S_n$. Следовательно, эти матрицы могут рассматриваться как точки в линейном пространстве размерности 20×L. Таким образом, на шаге инициализации требуется сгенерировать начальную популяцию организмов такую, чтобы каждый из порожденных организмов отстоял от других больше чем на некое изначально заданное расстояние D_0. Расстояние между двумя организмами M_1 и M_2 в таком случае вычисляется как обычное евклидово расстояние в линейном пространстве 20×L.

$$D = \sqrt{\sum_{i=1}^{20}\sum_{j=1}^{L}(m_1(i,j) - m_2(i,j))^2}$$

Экспериментально было выяснено, что наилучшая сходимость генетического алгоритма достигается в случае, когда заданное расстояние D_0 такое, что из изначально случайно сгенерированных 5×10^5 организмов оставалось лишь 1×10^4 – 1.05×10^4 удовлетворяющих заданному условию. Предел популяции в 1×10^4 организмов был выбран таким, поскольку, как было выяснено экспериментально, именно такое число матриц достаточно велико, чтобы максимально полно покрывать все пространство 20×L и вполне достаточно для того, чтобы алгоритм как можно реже сходился к локальным максимумам. Для генерации случайных организмов, был использован следующий алгоритм:

1. Построить искусственную последовательность S^*, образованную конкатенацией входных последовательностей $S_1, \dots, S_n$.
2. Перемешать случайным образом последовательность $S^* \rightarrow S^{*\prime}$
3. Для полученной последовательности $S^{*\prime}$ рассчитать позиционную матрицу – O

Для расчёта по последовательности длины n×L позиционно весовой матрицы O данная последовательность заново разбивалась на n отдельных последовательностей $S'_1, \dots, S'_n$. После чего для данных последовательностей рассчитывалась матрица частот – $M^{20\times L}$. Далее по строкам и столбцам данной матрицы соответственно рассчитывались значения:

$$x_1, \dots, x_L \text{ и } y_1, \dots, y_{20}$$

$$x_i = \sum_{j=1}^{20} M_{i,j}$$

$$y_i = \sum_{j=1}^{L} M_{i,j}$$

Далее по полученным для матрицы значениям $x_1, \dots, x_L$ и $y_1, \dots, y_{20}$ рассчитывались значения a,p,σ как:

$$a_{i,j} = \frac{x_i y_j}{nL}$$

$$p_{i,j} = \frac{x_i y_j}{(nL)^2}$$

$$\sigma_{i,j} = \sqrt{nLp_{i,j}(1 - p_{i,j})}$$

Итоговая случайная позиционно весовая матрица организма O вычислялась как

$$O_{i,j} = \frac{m_{i,j} - a_{i,j}}{\sigma_{i,j}}$$

Таким образом, общий алгоритм генерации начального множества организмов (начального поколения) Q_0 имеет следующий вид.

1. Сгенерировать очередной случайный организм O
2. Если $Q_0 = \{\emptyset\}$, то добавить организм O к множеству Q_0, иначе перейти к шагу 3
3. Если $\forall j$ такого, что $q_j \in Q$, выполнено $D(O, q_j) > D_0$, добавить организм O к множеству Q_0, иначе переход к шагу 1.

Модификация

Для каждой матрицы организма в данной работе можно рассчитать соответствующие ей значения K_d и R по приведенным ниже формулам:

$$R^2 = \sum_{i=1}^{20}\sum_{j=1}^{L}(m(i,j))^2$$

$$K = \sum_{i=1}^{20}\sum_{j=1}^{L} m(i,j)p(i,j)$$

где $p(i,j) = \frac{1}{20}(\frac{b(i)}{nL})$, $b(i)$ – кол-во вхождений аминокислоты i в последовательность S^* длины n×L, здесь n – кол-во последовательностей, а L – длина каждой из них.

Для того, чтобы проводить выравнивание нужно чтобы данные коэффициенты(R и K) были одинаковы для каждой матрицы. Это накладывает на матрицы условие, которое заключается в том, что функции распределения значений F_{sim} должны быть максимально похожими для всех матриц. Очевидно, что изначально заданные матрицы(организмы), такому условию не удовлетворяют, поэтому перед тем как рассчитывать, для них значение функции приспособленности F_{sim}, нужно провести модификацию матриц.

Процесс получения модифицированной матрицы организма – M' из изначальной матрицы организма – M , состоит в простом расчёте значений модифицированной матрицы – $m'_{i,j}$ по значениям исходной матрицы – $m_{i,j}$ используя формулу:

$$m'(i,j) = bp(i,j) + t\big(m(i,j) + p(i,j)(a-b)\big) \qquad (*)$$

Фигурирующие в этой формуле значения a, b и t рассчитываются следующим образом:

$$a = \frac{K_d - K_0}{\sum_{i=1}^{20}\sum_{j=1}^{L}(p(i,j))^2}$$

$$b = \frac{K_d}{\sum_{i=1}^{20}\sum_{j=1}^{L}(p(i,j))^2}$$

Где K_d – имеет значение параметра K, которое мы хотим видеть по итогу модификации для всех матриц организмов, а K_0 – это начальное значение K для не модифицированной матрицы M.

$$t_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

Из двух значений $t_{1,2}$ правильным считается то, которое позволяет получить модифицированную матрицу M'(по формуле (*)), которая будет ближе всего (по введенной евклидовой метрике) к исходной матрице M.

Коэффициенты A, B и C, в свою очередь могут быть рассчитаны по формулам приведенным ниже

$$A = 1$$

$$B = \frac{\sum_{i=1}^{20}\sum_{j=1}^{L} 2bp(i,j)\big(m(i,j) + p(i,j)(a-b)\big)}{\sum_{i=1}^{20}\sum_{j=1}^{L}(m(i,j) + p(i,j)(a-b))^2}$$

$$C = \frac{\sum_{i=1}^{20}\sum_{j=1}^{L}(bp(i,j))^2 - R^2}{\sum_{i=1}^{20}\sum_{j=1}^{L}(m(i,j) + p(i,j)(a-b))^2}$$

Процесс модификации матриц принимает в качестве **входа** саму матрицу организма M, рассчитанное для неё(по формуле для K) значение K_0, значения b(i) для каждой из 20 аминокислот соответственно, а также итоговые значения R и K_d для модифицированной матрицы M'.

На выходе получается модифицированная матрица M', с заданными на вход итоговыми значениями R и K_d.

Расчёт целевого значения функции

В качестве целевой функции для всех особей выступала функция F_{sim}. Эта функция принимает значения веса множественного глобального выравнивания, построенного с помощью заданной позиционно весовой матрицы(организма).

$$F_{sim}:\ q_i \to R \text{ ,где } q_i \in Q$$

Таким образом, для того чтобы вычислить целевую функцию для организма, требуется построить глобальное множественное выравнивание всех последовательностей с данной позиционно весовой матрицей, с афинной функцией штрафа за делецию. Строилось данное выравнивание, с помощью методов динамического программирования. Последовательности $S_1,\ldots,S_n$ объединялись в одну последовательность S^*, длина итоговой последовательности S^* равна соответственно n×L. Далее, с помощью позиционно весовой матрицы организма – W, строятся 3 матрицы выравнивания $M^{nL\times nL}$, $I_x^{nL\times nL}$, $I_y^{nL\times nL}$ по следующим рекуррентным соотношениям:

$$i = 1,\ldots, nL\text{ -}1;\ j = 1,\ldots,nL\text{ -}1$$

$$M(i,j) = \max\begin{cases} I_y(i-1,j-1) + W^T(i\ mod\ L, g(j)) \\ I_x(i-1,j-1) + W^T(i\ mod\ L, g(j)) \\ M(i-1,j-1) + W^T(i\ mod\ L, g(j)) \end{cases}$$

$$I_x(i,j) = \max\begin{cases} M(i-1,j) - d \\ I_x(i-1,j) - e \end{cases}$$

$$I_y(i,j) = \max\begin{cases} M(i,j-1) - d \\ I_y(i,j-1) - e \end{cases}$$

где d – штраф за открытие разрыва, e – штраф за продолжение разрыва, g(x) – функция, которая номеру столбца матрицы W, ставит в соответствие одну из 20 возможных аминокислот. Итоговым значением функции F_{sim} является значение веса выравнивания в правом верхнем углу матрицы M.

$$F_{sim} = M(nL, nL)$$

Начальными условиями и граничными условиями в таком случае являются:

$$\begin{cases} M(0,0) = I_x(0,0) = I_y(0,0) = 0 \\ M(0,j) = I_x(0,j) = I_y(0,j) = -10^6, \text{при } j \neq 0 \\ M(i,0) = I_x(i,0) = I_y(i,0) = -10^6, \text{при } i \neq 0 \end{cases}$$

Для того чтобы по полученной позиционно весовой матрице M восстановить множественное выравнивание требуется пройтись по найденному пути максимального выравнивания в матрице, восстанавливая каждый переход также как это делается, например в алгоритме Нидлмана- Вунша. Итогом будет две последовательности, в одна из которых будет состоять целиком из номеров столбцов позиционно весовой матрицы, либо пропусков(*), а вторая из аминокислот, либо пропусков(*).

Пример при L = 3:

```
12312*31*23
a*tctcactat
```

Итоговое множественное выравнивание:

```
1*2*3
a***t
c*tca
cta*t
```

Таким образом, по двум заданным последовательностям, можно однозначно восстановить множественное выравнивание последовательностей, если воспринимать первую последовательность, как позиции элемента из второй последовательности в итоговом множественном выравнивании.

Селекция

Под селекцией (или отбором) - подразумевается этап генетического алгоритма, на котором выбирается определенная доля от всей текущей популяции и только эта доля остается «в живых» и производит наследников на дальнейших этапах. В данной работе селекция проводилась методом рулетки. Для этого сначала все организмы отсортировывались в убывающем порядке по значению функции сходства F_{sim}. После чего из них выбиралось 20% самых приспособленных, для участия в дальнейшем размножении. Порог отсечения в 20% был выбран, основываясь на результате проведенных экспериментов, уменьшение процента выбранных особей, приводило к большей вероятности достигнуть локального максимума при выполнении генетического алгоритма.

Пусть по результатам предыдущего этапа размер «выживших» в результате селекции особей популяции равен K. Выбор двух конкретных родителей из множества оставшихся «в живых» особей производился по методу рулетки, который базируется на следующем принципе. Вероятность размножения конкретного организма тем выше, чем выше его значение функции сходства F_{sim}. Таким образом, пусть F_{sim}^{i} значение функции сходства для i-го организма, тогда вероятность выбрать i-ый организм в качестве родителя – P_i, рассчитывается как:

$$P_i = \frac{F_{sim}^{i}}{\sum_{j=1}^{K} F_{sim}^{j}}$$

Такой метод отбора позволяет добиться того, что менее приспособленные особи будут тоже иметь возможность для размножения. Такой процесс отбора нужен именно для того, чтобы популяция не вырождалась быстро, и алгоритм сходился к глобальному, а не локальному максимуму.

Скрещивание

Для скрещивания (или размножения) организмов использовалась комбинация методов двухточечного кроссовера и дифференциального перехода. Метод двухточечного кроссовера заключался в следующем. Предположим, что мы выбрали(методом рулетки) для скрещивания два организма – M_1 и M_2 . В таком случае, для того чтобы провести между ними скрещивание, представим соответствующие M_1 и M_2 матрицы в виде вектора. Это можно довольно просто сделать, если просто записать все строки матрицы последовательно друг за другом в одной последовательности. Получив два таких вектора, мы можем замкнуть их в круг, «склеив» их концы и получив из них два кольца. Затем на кольце случайным образом выбираются две точки, и участок, оказавшийся между ними, вырезается из кольца и заменяется на аналогичный участок из другого кольца. Применение кроссовера более высокого порядка в таком случае, является нецелесообразным, поскольку это увеличивает «разрушаемость» организмов и замедляет эволюционные процесс генетического алгоритма.

После того как было получено два новых организма, которые обладают свойством «похожести» на родителей, применяется дифференциальный переход. Этот метод позволяет с некоторой вероятностью получить организмы с более высоким значением функции сходства, чем у родителей. Дифференциальный переход осуществляется следующим образом: пусть M_1' и M_2' - организмы, полученные после применения двухточечного кроссовера. В таком случае значения элементов матриц новых организмов M_1'' и M_2'' могут быть рассчитаны по следующим формулам:

$$i = 1,\ldots,20\ ;\ j = 1,\ldots,L$$

$$m_1''(i,j) = \alpha m_1'(i,j) + (1-\alpha)m_2'(i,j)$$

$$m_2''(i,j) = (1-\alpha)m_1'(i,j) + \alpha m_2'(i,j)$$

Здесь α – случайная величина, имеющая равномерное распределение на отрезке [0,1]

Мутация

Мутации над организмами производились при помощи одного из двух возможных методов. Какой метод будет первым, выбиралось случайно. Первый метод заменяет случайно выбранный из матрицы элемент, на случайное число равномерно распределенное в диапазоне от [-1,1]. Вероятность замены при этом равна p_1 = 0.01. Второй метод изменял элементы всей матрицы на некоторое небольшое значение. Вероятность такой замены p_2 была в пределах между 0.001 и 0.03. Таким образом каждый элемент матрицы M – $m(i,j)$, заменялся на новый элемент $m'(i,j)$ по следующей формуле

$$i = 1,\ldots,20\ ;\ j = 1,\ldots,L$$

$$m'(i,j) = m(i,j) \pm p_2 m(i,j)$$

Вероятности p_1 и p_2 были подобранны экспериментально. Выбор конкретных значений p_1 и p_2 в данном случае зависит от двух факторов. Требуется, чтобы популяция в процессе эволюции, и генетический алгоритм в целом не сходились к локальному максимуму. Также новые организмы не должны быть полностью случайными, они должны быть близки к уже существующим организмам, потому что в противном случае результат не будет улучшаться.

Статистическая мера значимости

Мера значимости всего построенного алгоритмом выравнивания – Z рассчитывается в конце работы генетического алгоритма для итогового организма (организма с наибольшим значением функции F_{sim}). Для проведения расчета k раз строится глобальное выравнивание случайно перемешанной последовательности S^*, с итоговой матрицей(организмом) $\widetilde{M}$. После чего рассчитываются оценки математического ожидания и дисперсии величины F_{r_sim} для случайных последовательностей. Затем значение величины Z рассчитывалось по формуле

$$Z = \frac{F_{sim} - \widetilde{M}(F_{r\_sim})}{\sqrt{\widetilde{D}(F_{r\_sim})}}$$

В случае если полученное значение Z больше чем выбранный экспериментально порог Z_0, то можно считать, что получено осмысленное выравнивание последовательностей $S_1, \dots, S_n$.