Strijov V.V., Kuznetsov M.P., Rudakov K.V. Rank-scaled metric clustering of amino-acid sequences // Mathematical Biology and Bioinformatics, 2012, 7(1) : 345-359.

Математическая биология и биоинформатика. 2012. Т. 7. № 1. С. 345–359.
URL: http://www.matbio.org/2012/Strijov2012(7_345).pdf

========== ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ ==========

УДК 519.95

Метрическая кластеризация последовательностей аминокислотных остатков в ранговых шкалах[1]

©2012 Стрижов В.В.[*1], Кузнецов М.П.[2], Рудаков К.В.[1]**

[1]*Вычислительный центр им А.А. Дородницына, Российская академия наук, Москва, 19333, Россия*
[2]*Московский физико-технический институт (Государственный университет), Долгопрудный, Московская область, 141700, Россия*

Аннотация. Для решения задачи распознавания вторичной структуры белков предложен алгоритм кластеризации подпоследовательности аминокислотных остатков. Для выявления кластеров используются парные расстояния между подпоследовательностями. Отличительной особенностью алгоритма является то, что не требуется строить полную матрицу парных расстояний, что снижает сложность вычислений. При кластеризации рассматриваются только ранги расстояний между подпоследовательностями. Работа алгоритма проиллюстрирована синтетическими данными и данными из базы UniProtKB.

Ключевые слова: *кластеризация, функция расстояния, матрица парных расстояний, метрическая конфигурация, ранговые шкалы.*

ВВЕДЕНИЕ

Предлагаемый алгоритм был разработан в ходе решения задачи прогнозирования вторичной структуры белка по первичной [1,2]. Предлагалось найти соответствие между «типичными» последовательностями аминокислотных остатков, кодируемых буквами двадцатибуквенного алфавита, и соответствующими им вторичными структурами, кодируемыми буквами трехбуквенного алфавита. Для этого предполагалось составить словарь часто встречающихся «типичных» последовательностей аминокислотных остатков, то есть решить задачу кластеризации. Особенностью задачи является то, что база данных остатков, подпоследовательности которых требуется кластеризовать, содержит 11 миллионов записей длиной от 20 до 33000 символов каждая [3,4]. Такой объем данных выдвигает ограничение на сложность алгоритма кластеризации; предполагается возможность параллельного запуска этого алгоритма.

Ранее были предложены алгоритмы быстрой кластеризации объектов, описанных в категориальных шкалах [5,6], на основе алгоритма *k*-means [7,8]. Данный алгоритм был принят в этой работе в качестве базового для сравнения [9].

Основная идея предложенного похода заключается в следующем. Каждая последовательность аминокислотных остатков разбивается на слова одинаковой длины.

[1]*Работа выполнена при поддержке РФФИ, грант 10-07-00422-а, и Минобрнауки России, контракт №07.514.11.4001.*
[*]strijov@ccas.ru
[**]mikhail.kuznecov@phystech.edu

Длина слова задается до начала кластеризации и выбирается исходя из результатов анализа записей о вторичных белковых структурах. Множество полученных слов является множеством кластеризуемых объектов. На этом множестве задана метрика, и далее его объекты будут называться точками, погруженными в метрическое пространство.

Вышеперечисленные алгоритмы кластеризации требуют матрицу парных расстояний между всеми парами точек из множества, что существенно повышает сложность алгоритма. Предложенный алгоритм требует только расстояния между выделенными точками, называемыми далее ρ-сетью [10], и всеми остальными точками. При этом расстояния от некоторой выделенной точки до прочих ранжируются, и кластеризация выполняется по ранговым значениям. Таким образом, алгоритм состоит из следующих основных шагов:

- создание набора парных расстояний,
- задание опорного множества (ρ-сети),
- вычисление расстояния между некоторыми парами объектов,
- нахождение метрических сгущений, кластеризация.

Далее в работе описаны метрики, используемые при кластеризации последовательностей аминокислотных остатков, и их свойства, указан способ построения ρ-сети, описан предложенный алгоритм кластеризации и указана его сложность. Затем описан вычислительный эксперимент, который содержит описание данных, базового алгоритма и принятого функционала качества кластеризации. Работу завершает сравнение и анализ результатов работы двух алгоритмов.

ФУНКЦИИ РАССТОЯНИЯ МЕЖДУ СЛОВАМИ

Опишем способ получения множества объектов кластеризуемой выборки. Задана цепочка букв двадцатибуквенного алфавита, $x_1,\ldots,x_i,\ldots,x_p$ длиной p, соответствующая первичной структуре некоторого белка. Множеством объектов будем считать множество $\{x_i,\ldots,x_{i+n-1} \mid i=0,\ldots,p-n-1\}$ слов заданной длины n. При наличии нескольких цепочек букв множества слов, полученные для каждой цепочки, объединяются. Представим каждое полученное слово в виде точки в пространстве. Такое представление позволяет погрузить $n+1$ точку в n-мерное пространство и, задав метрику между парами точек, найти наиболее близкие пары. Множества точек, имеющие относительно малые парные расстояния, будем называть *метрическим сгущением*.

Рассмотрим два слова: $\mathbf{x}=(x_1,...,x_n)$ и $\mathbf{y}=(y_1,...,y_m)$. В общем случае слова $\mathbf{x}$ и $\mathbf{y}$ могут быть разной длины. Необходимо, чтобы выбираемая нами функция расстояния между словами $\rho(\mathbf{x},\mathbf{y})$ была метрикой. Для этого должны быть выполнены следующие условия:

1. условие тождества, $\rho(\mathbf{x},\mathbf{y})=0 \Leftrightarrow \mathbf{x}=\mathbf{y}$;
2. условие симметрии, $\rho(\mathbf{x},\mathbf{y})=\rho(\mathbf{y},\mathbf{x})$;
3. неравенство треугольника $\rho(\mathbf{x},\mathbf{z}) \le \rho(\mathbf{x},\mathbf{y})+\rho(\mathbf{y},\mathbf{z})$.

Симметрическая разность на неупорядоченных множествах

Данная функция расстояния между словами $\mathbf{x}$ и $\mathbf{y}$ определена как

$$\rho(\mathbf{x},\mathbf{y})=\frac{|\mathbf{x}|+|\mathbf{y}|-2S(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|-S(\mathbf{x},\mathbf{y})},$$

где $S(\mathbf{x},\mathbf{y})$ – пересечение наборов $\mathbf{x}$ и $\mathbf{y}$ как неупорядоченных множеств: каждому элементу набора $\mathbf{x}$ ставится в соответствие тождественный ему элемент набора $\mathbf{y}$ без учета индексов последнего. Число полученных пар является значением функции S. При этом множества X, Y считаются неупорядоченными. Знак $|\cdot|$ означает мощность множества, в данном случае – число букв в слове. Элементы слов $\mathbf{x}$ и $\mathbf{y}$ индексированы, на множстве индексов задано отношение полного порядка.

Для данного расстояния, очевидно, выполнено условие симметрии. Также выполнено неравенство треугольника. Докажем его для случая $|\mathbf{x}|=|\mathbf{y}|=|\mathbf{z}|$, потому что предложенный алгоритм будет использовать только одинаковые длины слов. Обозначим

$$a=\frac{S(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|}, b=\frac{S(\mathbf{y},\mathbf{z})}{|\mathbf{y}|+|\mathbf{z}|}, c=\frac{S(\mathbf{x},\mathbf{z})}{|\mathbf{x}|+|\mathbf{z}|}.$$

Тогда

$$\rho(\mathbf{x},\mathbf{y})+\rho(\mathbf{y},\mathbf{z})-\rho(\mathbf{x},\mathbf{z})=\frac{|\mathbf{x}|+|\mathbf{y}|-2S(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|-S(\mathbf{x},\mathbf{y})}+\frac{|\mathbf{y}|+|\mathbf{z}|-2S(\mathbf{y},\mathbf{z})}{|\mathbf{y}|+|\mathbf{z}|-S(\mathbf{y},\mathbf{z})}-\frac{|\mathbf{x}|+|\mathbf{z}|-2S(\mathbf{x},\mathbf{z})}{|\mathbf{x}|+|\mathbf{z}|-S(\mathbf{x},\mathbf{z})}=$$

$$=1-\frac{S(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|-S(\mathbf{x},\mathbf{y})}-\frac{S(\mathbf{y},\mathbf{z})}{|\mathbf{y}|+|\mathbf{z}|-S(\mathbf{y},\mathbf{z})}+\frac{S(\mathbf{x},\mathbf{z})}{|\mathbf{x}|+|\mathbf{z}|-S(\mathbf{x},\mathbf{z})}=$$

$$=1-\frac{a}{1-a}-\frac{b}{1-b}+\frac{c}{1-c}=\frac{1-2a-2b+3ab+ac+bc-2abc}{(1-a)(1-b)(1-c)}.$$

Чтобы неравенство треугольника выполнялось, необходимо, чтобы эта дробь была неотрицательной. Поскольку все $a,b,c\in[0;1/2]$, знаменатель является положительным числом. Заметим также, что для a,b и c выполнено соотношение $c\geq a+b-1/2$. Это так, потому что наибольшее по мощности множество букв, состоящее из объединения пересечений слов $\mathbf{x},\mathbf{y}$ и $\mathbf{y},\mathbf{z}$, не содержащихся в пересечении $\mathbf{x},\mathbf{z}$, равно $|\mathbf{x}|$. Обозначим через u мощность этого объединения, через u' мощность множества букв, состоящего из объединения пересечений $\mathbf{x},\mathbf{y}$ и $\mathbf{y},\mathbf{z}$, содержащихся в пересечении $\mathbf{x},\mathbf{z}$. Тогда:

$$\frac{S(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|}+\frac{S(\mathbf{y},\mathbf{z})}{|\mathbf{y}|+|\mathbf{z}|}=\frac{u+u'}{2|\mathbf{x}|}\leq\frac{1}{2}+c.$$

Поэтому числитель дроби

$$1-2a-2b+3ab+ac+bc-2abc\geq 1-\frac{5}{2}a-\frac{5}{2}b+6ab+a^2+b^2-2a^2b-2ab^2.$$

Заметим, что эта дробь симметрична по a и b, поэтому для глобального минимума должно выполняться $a=b$. Симметризуя это выражение, получаем:

$$1-5a+8a^2-4a^3,$$

которое неотрицательно при $a\in[0,1/2]$. Значит, не отрицателен и числитель дроби, и неравенство треугольника в этом случае выполнено.

Симметрическая разность на упорядоченных множествах

Данная метрика определена как

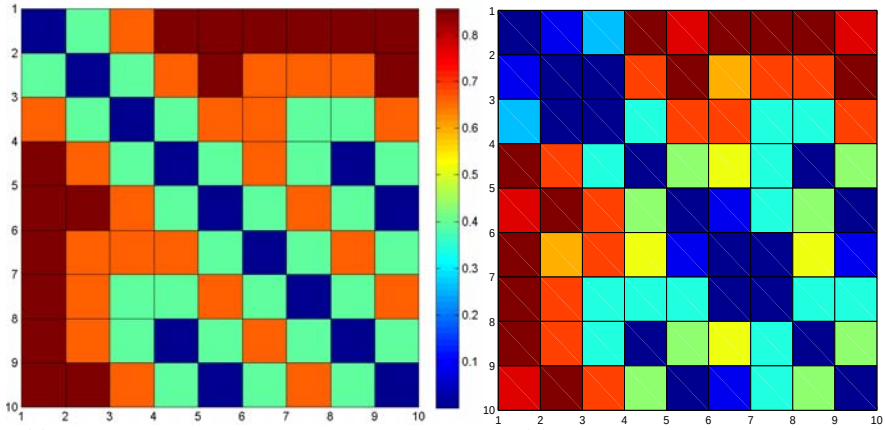
$$\rho(\mathbf{x},\mathbf{y}) = \frac{|\mathbf{x}|+|\mathbf{y}|-2G(\mathbf{x},\mathbf{y})}{|\mathbf{x}|+|\mathbf{y}|-G(\mathbf{x},\mathbf{y})},$$

где $G(\mathbf{x},\mathbf{y})$ – мощность наибольшей общей подпоследовательности символов в словах $\mathbf{x}$ и $\mathbf{y}$. Мощность пересечения двух упорядоченных наборов символов (наибольшей общей подпоследовательности) равна длине диагонального пути наименьшей стоимости, определенного в следующем разделе.

Данное расстояние является метрикой, потому что для него выполнены условие симметрии и неравенство треугольника (аналогично предыдущему случаю), а также верно условие тождества:

$$\rho(\mathbf{x},\mathbf{y}) = 0 \Leftrightarrow G(\mathbf{x},\mathbf{y}) = |\mathbf{x}| = |\mathbf{y}|,$$

а это возможно только в том случае, когда наибольшая общая подпоследовательность совпадает со всем словом, то есть два слова тождественны. Область значения данной функции расстояния находится на отрезке $[0;1]$. На рис. 1 слева показана матрица парных расстояний D для этой метрики. Каждый элемент матрицы есть значение функции расстояния для соответствующей пары слов.



(а) Симметрическая разность (б) Оптимальное выравнивание

Рис. 1. Матрица парных расстояний, пример.

Оптимальное выравнивание

Подсчет этого расстояния сводится к поиску оптимального выравнивания между двумя словами. Расстоянием между двумя буквами x_i, y_j этих слов является булева функция:

$$d_{i,j} = \begin{cases} 1, \text{если } x_i \neq y_j, \\ 0, \text{иначе.} \end{cases}$$

Для вычисления расстояния между словами составим $M(n+1 \times m+1)$-матрицу стоимости. Обозначим индекс первой строки $i = 0$ и индекс первого столбца $j = 0$. Присвоим

$$M(0,0) = 0;$$

для всех $i = 1,...,n$ и $j = 1,...,m$ присвоим

$$M(0,j) = M(i,0) = \infty;$$

для всех $i = 1,..., n$ и $j = 1,..., m$ вычислим последовательно все элементы матрицы M по формуле

$$M(i, j) = d(x_i, y_j) + \min(M(i-1, j-1), M(i-1, j), M(i, j-1)).$$

Искомым расстоянием между словами $\mathbf{x}$ и $\mathbf{y}$ будет последний элемент этой матрицы:

$$\rho(\mathbf{x}, \mathbf{y}) = M(n, m). \quad (1)$$

Стоит отметить, что данное расстояние является частным случаем расстояния Левенштейна [11], то есть является метрикой. На рис. 1 справа показана матрица парных расстояний для случая $\rho(\mathbf{x}, \mathbf{y}) \in [0,1]$, длина слов $m = n = 8$. На рис. 2 показана матрица стоимости M алгоритма оптимального выравнивания. Путь наименьшей стоимости показан точками. Его начало и конец фиксированы в элементах с индексами (0,0) и (n, m).

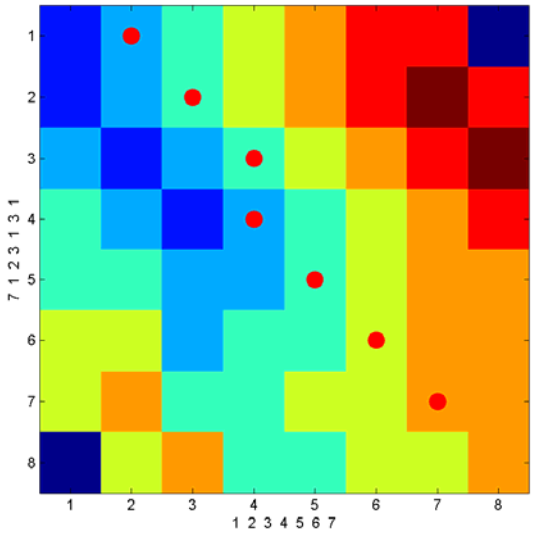


Рис. 2. Матрица стоимости оптимального выравнивания.

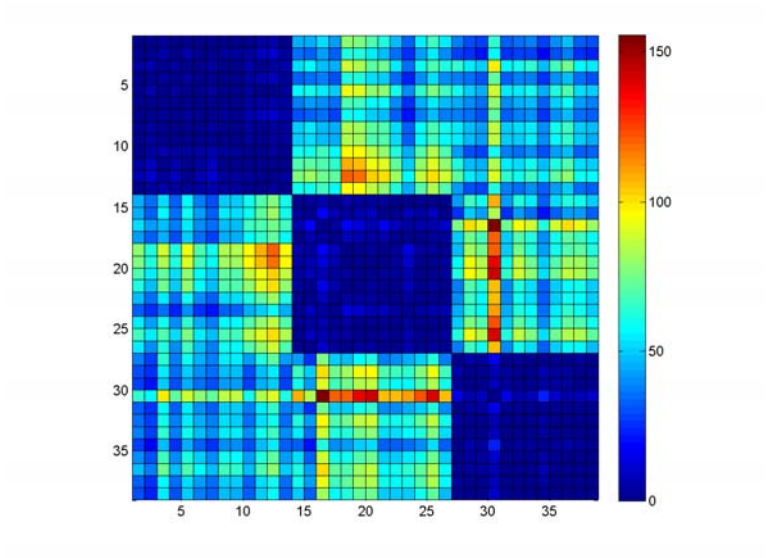


Рис. 3. Матрица парных расстояний для 40 точек.

АЛГОРИТМ КЛАСТЕРИЗАЦИИ НА ОСНОВЕ ρ-СЕТЕЙ

Опишем алгоритм, позволяющий быстро кластеризовать объекты в произвольном метрическом пространстве. На рис. 3 показана симметричная относительно главной диагонали матрица парных расстояний для 40 точек. При создании нижеописанного алгоритма преследовалась цель существенно снизить сложность процедуры кластеризации относительно квадратичной, требуемой для построения матрицы парных расстояний между всеми объектами кластеризуемого множества.

Обозначим $X = \{\mathbf{x}_1,...,\mathbf{x}_N\}$ – множество, состоящее из N точек. Задана функция расстояния $\rho(\mathbf{x}_i,\mathbf{x}_j)$, определенная на всех парах точек из X, для которой выполняются условия метрики. Требуется найти множество $K \subset X$ – подмножество X, образующее метрическое сгущение. Сгущением называется множество близких, в смысле заданной метрики, точек, образующих компактные области. Считается, что множеству точек, образующих сгущение, принадлежат все точки выпуклой комбинации этого множества. Предполагается, что будет найдена последовательность сгущений K посредством итеративной процедуры следующего вида. Из заданного набора X вычитаем множество точек K, образующих сгущение, $X^* = X \setminus K$. Находим сгущение K^* на полученном наборе X^*. Процедура повторяется до нахождения всех сгущений $\{K\}$.

Для отыскания множества K введем понятие ρ-сети и построим матрицу D парных расстояний между точками, принадлежащими ρ-сети, и всеми точками множества X: $D = \{d_{i,j}\}$, где $i \in \{1,...,n\} = I$ – индекс объекта ρ-сети, а $j \in \{1,...,N\} = J$ – индекс объекта из X.

ρ-сеть – это множество $X' = \{\mathbf{x}_k \mid k \in I\}$ фиксированной мощности n, собственное подмножество X, состоящее из объектов, которые находятся на максимальном расстоянии друг от друга, т. е.

$$I = \arg\max_{j\in J}\min_{i\in I/\{j\}} \rho(\mathbf{x}_i,\mathbf{x}_j), \quad I \subset J.$$

Точки, входящие в ρ-сеть X', также принадлежат множеству X, $X' \subset X$, причем предполагается, что $N = |X| \gg n = |X'|$. Множество точек ρ-сети отыскивается с помощью следующей процедуры.

Выбор точек для ρ-сети

Положим изначально $X' = \varnothing$ – множество точек ρ-сети.

1. Взять произвольный элемент $\mathbf{y} \in X$.
2. Вычислить $\mathbf{x}' = \arg\max_{\mathbf{x}\in X} \rho(\mathbf{x},\mathbf{y})$, присвоить $X' := X' \cup \mathbf{x}'$.
3. Пока $|X'| < n$: вычислить $\mathbf{x}' = \arg\max_{\mathbf{x}\in X}\min_{\mathbf{z}\in X'} \rho(\mathbf{x},\mathbf{z})$, присвоить $X' := X' \cup \mathbf{x}'$.

Отметим, что предложенный алгоритм имеет сложность $O(n^2 N)$, где $n^2 \ll N$, то есть линейную по числу объектов.

Построение матрицы парных расстояний

Построим матрицу D парных расстояний между точками, принадлежащими ρ-сети, и всеми точками из X: $D = \{d_{ij}\}, d_{ij} = \rho(\mathbf{x}_i,\mathbf{x}_j)$, где $i \in I$ – индекс объекта сети, а $j \in J$ –

индекс объекта из X. Матрица $D \in \mathsf{R}_+^{n\times N}$ содержит в своих строках расстояния от каждого объекта ρ-сети до каждого объекта из всего множества X.

Сортировка матрицы парных расстояний

Для каждой строки i матрицы D зададим функцию φ_i, которая индексам элементов строки ставит в соответствие индексы отсортированных по возрастанию расстояний от i-й точки ρ-сети до всех точек множества X:

$$\varphi_i : \{\rho_{ij} \mid j \in J\} \mapsto \{\mathrm{sort}(\rho_{ik}) \mid k \in J\}.$$

Функция φ_i задает преобразование $J \to J$ – биекцию

$$\varphi_i : j \mapsto k, \quad j,k \in I.$$

Построим матрицу, содержащую в строках индексы $r_{ij} \in \mathbb{N}$ отсортированных значений расстояний

$$R = \{r_{ij} \mid r_{ij} = \varphi_i(j)\}$$

и индексы $r'_{ij} \in \mathbb{N}$ обратных относительно операции сортировки значений

$$R' = \{r'_{ik} \mid r'_{ik} = \varphi_i^{-1}(k)\}.$$

Другими словами, матрица $R \in \mathsf{N}^{n\times N}$ содержит в своих строках индексы расстояний от i-й точки ρ-сети до j-й точки из множества X, отсортированных по возрастанию. Пример для фиксированного i и $j \in \{1,2,3\}$ показан в табл. 1.

Табл. 1: Пример строки матрицы парных расстояний и соответствующих ранговых значений

Индексы точек	1	2	3
ρ_{ij}	0.7	0.3	0.5
sort(ρ_{ij})	0.3	0.5	0.7
r_{ij}	3	1	2
r'_{ij}	2	3	1

Поиск метрического сгущения

На строках матрицы R' зададим окно заданной ширины, включающее $w = \lfloor (1/2)k \cdot N \rfloor$ элементов строки. Здесь k – задаваемый параметр, описывающий выраженность сгущения. За центр окна примем k-й столбец матрицы R', индекс $k \in \{w+1,..., N-w-1\}$.

Найдем кластер K с наибольшим количеством элементов, $|K| \to \max$. Для этого для каждого номера точки $j \in J$ в каждой строке с номером i матрицы R' найдем окрестность $K_i \subset J$, соседние элементы j-го столбца, мощностью $2w+1$. Кластером K будет являться пересечение множеств K_i по всем i:

$$K = \bigcap_{i=1}^{n} K_i.$$

Опишем процедуру поиска сгущения. Примем изначально $K := J$. Далее

1. для всех индексов точек множества X $j \in J$ и для всех индексов точек ρ-сети $i \in I$:
2. найти ближайших соседей K_i точки $\mathbf{x}_j \in X$ относительно точки ρ-сети $\mathbf{x}_i \in X'$:

$$K_i = \{r'_{is} : s \in \{r_{ij} - w, ..., r_{ij} + w\}\}. \quad (2)$$

3. Присвоить $K := K \cap K_j$.

Примечание: на шаге 2) алгоритма возникнет ситуация, когда

$$r_{ij} - w < 0 \quad \vee \quad r_{ij} + w > N.$$

В первом случае, надо брать $s:\ s \in \{1, ..., 2w+1\}$, а во втором $s:\ s \in \{N - 2w, ..., N\}$.

Сложность алгоритма

Предлагаемый алгоритм имеет сложность $O(n^2 N)$ при построении матрицы расстояний D, $O(nN \log N)$ при сортировке строк матрицы D и $O(nN)$ при поиске метрических сгущений. На рис. 4 показана сложность предложенного алгоритма, в сравнении со сложностью алгоритма *k*-means.

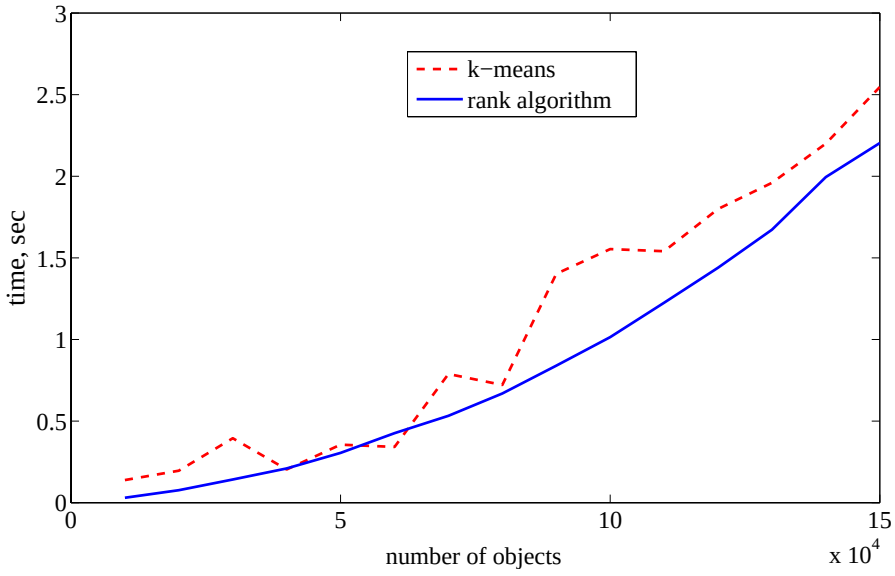


Рис. 4. Сложность алгоритма поиска сгущения относительно количества слов.

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Вышеописанный алгоритм кластеризации тестировался на кластерах белков из базы данных UniRef [12] первичных структур белков UniprotKB [13] и сгенерированных нами синтетических данных.

База данных кластеров белков UniRef

База знаний о последовательностях белков UniProtKB содержит все известные первичные структуры белков. База данных UniRef, являющаяся частью UniProtKB, содержит кластеры белков, построенные с учетом сходства первичных структур. Мы использовали записи базы в формате FASTA Sequence data [3]. Описание формата FASTA представления кластеров белковых первичных структур приведено на примере одной записи [4]:

```
>sp|Q08753|TRHN1_CHLMO Group 1 truncated hemoglobin LI637 OS=Chlamydomonas moewusii
GN=LI637 PE=1 SV=1
MMRTVQLRTLRPCIRAQQQPVRPSTSATAAAATAPAPARKCPSSLFAKLGGREAVEAAVD
KFYNKIVADPTVSTYFSNTDMKVQRSKQFAFLAYALGGASEWKGKDMRTAHKDLVPHLSD
VHFQAVARHLSDTLTELGVPPEDITDAMAVVASTRTEVLNMPQQ
```

Здесь с третьей строки начинается последовательность аминокислот, которая называется представителем кластера (она выбирается из всех последовательностей, входящих в кластер, по определенным правилам). Строка, начинающаяся с символа “>”, является описанием кластера. Она содержит фиксированные поля описания конкретного кластера, имеет неопределенную длину и всегда предваряет последовательность. Аминокислотная последовательность записывается в последующих строках длиной, не превышающей 60 символов в строке. Индикатором конца последовательности является появление либо символа “>”, означающего начало нового кластера, либо конца файла.

При разбиении цепочки аминокислот использовано окно с фиксированной шириной. При каждом запуске вычислительного эксперимента ширина окна назначалась в промежутке от 5 до 30. Данные значения обусловлены анализом длины типичных слов в цепочках вторичных структур восьмибуквенного словаря DSSP [14]. Длина шага (количество позиций, на которое сдвигалось окно вдоль последовательности) равнялась 1.

Результат сегментации показан на примере аминокислотной последовательности длиной 12 символов MVLSEGEWQLVL. После разбиения с длиной окна 7 цепочка имеет вид: MVLSEGE, VLSEGEW, LSEGEWQ, SEGEWQL, EGEWQLV, GEWQLVL.

Выборка синтетических кластеров

При создании синтетической выборки порождались нормально-распределенные множества точек со следующими параметрами: количество порожденных кластеров, количество кластеризуемых, точек среднее расстояние между кластерами, разброс внутри кластеров. Предполагалось, что в среднем в каждом кластере равное число точек.

Функция ошибки кластеризации

Для оценки качества кластеризации была введена следующая функция ошибки. Среднее внутрикластерное расстояние должно быть как можно меньше:

$$F_0 = \frac{\sum\limits_{i<j} [k_i = k_j] \rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum\limits_{i<j} [k_i = k_j]} \to \min, \quad i, j \in \{i, \ldots, N\}.$$

Здесь индикаторная функция $[k_i = k_j]$ означает, что если точки с индексами i и j принадлежат одному и тому же кластеру с номером k, то возвращается единица, в противном случае – ноль. Среднее межкластерное расстояние должно быть как можно

больше:

$$F_1 = \frac{\sum_{i<j}[k_i \neq k_j]\rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{i<j}[\mathbf{y}_i \neq \mathbf{y}_j]} \to \max \quad i, j \in \{i, \ldots, N\}$$

Зададим функцию ошибки кластеризации как отношение среднего внутрикластерного и среднего межкластерного расстояния:

$$Q = \frac{F_0}{F_1} \to \min.$$

Результатом работы алгоритма кластеризации должен быть кластер максимальной мощности, содержащий слова максимальной длины.

Базовый алгоритм

В качестве базового алгоритма, с которым сравнивался предложенный, был выбран алгоритм «*k* средних», или *k*-means [7,8]. В качестве параметра алгоритм принимает на вход количество кластеров, а на первом шаге делает начальное приближение центров кластеров, которые затем итеративно пересчитывает.

Алгоритму также необходимо признаковое описание объекта: набор функций $f_j : X \to \mathsf{R}, j = 1,..., m$, где m – количество признаков. В случае точек на плоскости, признаковое описание состоит из координат точек. Алгоритм выполняется следующим образом.

1. Сформировать начальное приближение центров всех кластеров: $\mu_k, k = 1,..., K$.
2. Отнести каждый объект к ближайшему центру:

$$k_i := \arg\min_{k \in K} \rho(\mathbf{x}_i, \mu_k), i = 1,..., N,$$

3. Вычислить новое положение центров:

$$\mu_{kj} := \frac{\sum_{i=1}^{N}[k_i = k] f_j(\mathbf{x}_i)}{\sum_{i=1}^{N}[k_i = k]},$$

4. Повторять шаги 2, 3 пока значения k_i не перестанут изменяться.

Во второй формуле используется признаковое описание точек, функция f_j – j-е значение вектора описания точки. В случае, когда используется только матрица парных расстояний, j-м признаком i-й точки считается соответствующий элемент i-й строки матрицы парных расстояний $f_j(\mathbf{x}_i) = \rho(\mathbf{x}_i, \mathbf{x}_j)$, $j \in \{1, \ldots, N\}$, в которой строка – набор расстояний от этой точки до всех остальных.

Сравнение работы двух алгоритмов кластеризации

Приведем сначала результаты визуального сравнения на синтетической выборке, а затем опишем результаты кластеризации аминокислотных последовательностей. На рис. 5 показаны результаты кластеризации. Алгоритм *k*-means, получив в качестве входного

параметра число кластеров, при «неудачном» начальном приближении центров кластеров разбил один порожденный кластер на две части, а два оставшихся объединил, см. рис 5 а). Предложенный алгоритм не получал число кластеров в качестве входного параметра и выявил четыре кластера, объединив последние два, что для решения рассматриваемой прикладной задачи является корректным результатом.

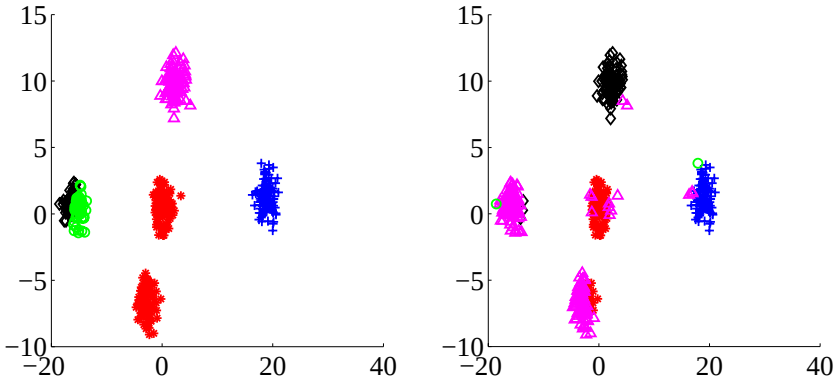


Рис. 5. Визуальное сравнение работы алгоритма *k*-means и алгоритма ранговой кластеризации.

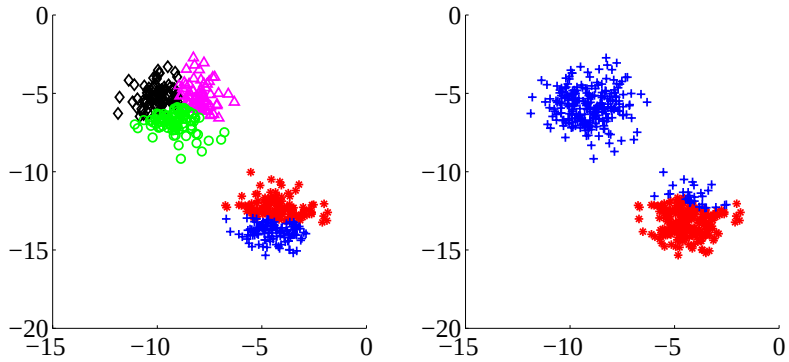


Рис. 6. Визуальное сравнение работы алгоритма *k*-means и алгоритма ранговой кластеризации.

На рис. 6 показаны два сгущения точек. Так как в алгоритме *k*-means, в отличие от рангового, в качестве параметра задано число кластеров, то кластеры были обнаружены некорректно, см. рис. 6 а). Ранговый алгоритм на рис. 6 обнаружил два кластера с удовлетворительной ошибкой.

На рис. 7 графике показаны два кластера, содержащие по 250 точек. У одного из кластеров разброс значений значительно меньше, чем у второго, и геометрически он целиком лежит внутри второго. Алгоритм *k*-means не может корректно отделить такие кластеры, это показано на рис. 7 а). В данном случае алгоритм поместил в один кластер 375, а в другой 125 точек. Алгоритм ранговой кластеризации гораздо лучше выделил сгущение , см. рис. 7 b), получив в результате 274 точки в одном кластере и 226 в другом.

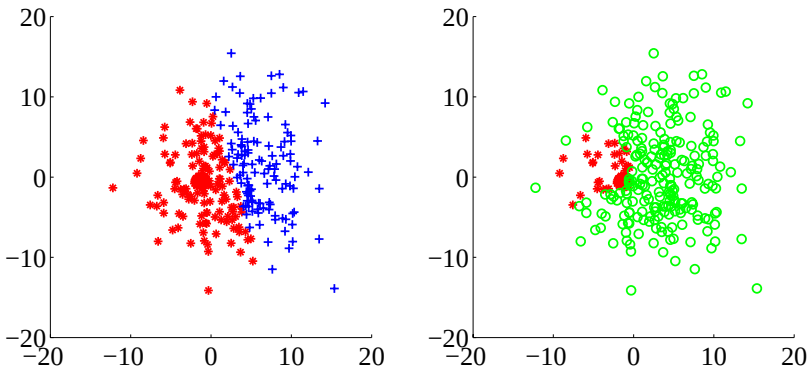


Рис. 7. Визуальное сравнение работы алгоритмов, случай вложенных кластеров.

Эмпирическая оценка оптимального значения k

На рис. 8 показана зависимость функции ошибки кластеризации Q от мощности ρ-сети n и параметра кластеризации k. Исходная выборка состоит из 500 точек, которые объединены в пять кластеров, как показано на рис. 5. Видно, что при значении $k > 0.5$ качество резко ухудшается. Это связано с тем, что алгоритм выбирает очень много точек в первые кластеры, и внутрикластерное расстояние становится большим. При маленьких значениях k ошибка кластеризации заметно меньше, потому что алгоритм отбирает небольшие, но выраженные сгущения. При увеличении n качество кластеризации незначительно увеличивается.

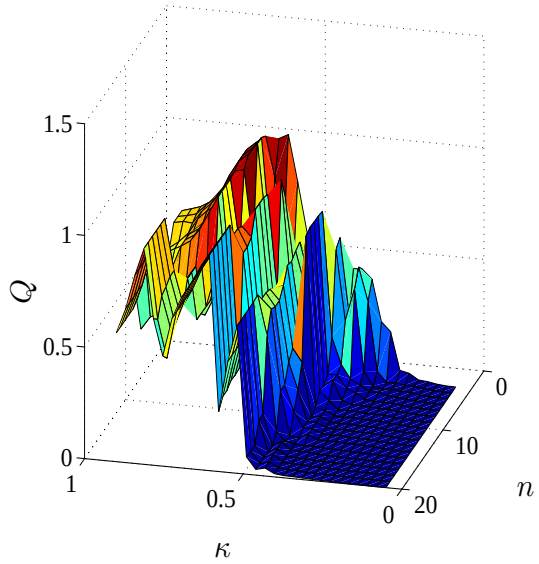


Рис. 8. Зависимость функции ошибки Q от мощности ρ-сети n и параметра кластеризации k.

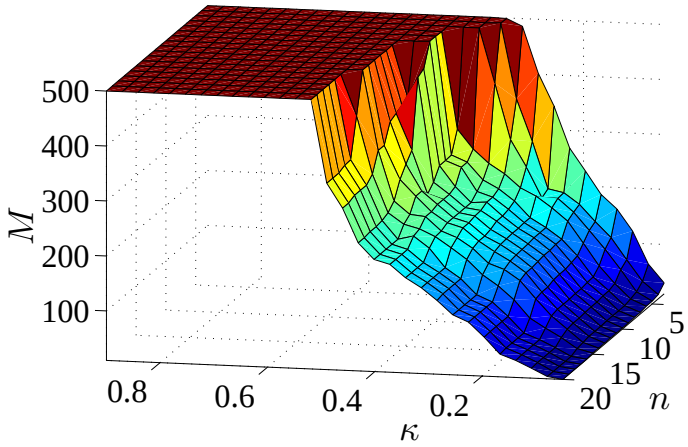


Рис. 9. Зависимость количества кластеризованных точек M от мощности ρ-сети n и параметра кластеризации k .

Однако для оценки качества предложенного алгоритма недостаточно использовать только функцию ошибки. На каждой итерации алгоритм выбирает кластер определенной мощности, значение которой пропорционально k. Таким образом, при завершении алгоритма кластеризуются не все точки выборки, а только некоторые M из них. На рис. 9 показана зависимость количества кластеризованных точек M от мощности ρ-сети n и параметра кластеризации k. Видно, что при тех значениях параметров n, k, при которых функция ошибки принимает наилучшее значение, кластеризуются не все точки выборки. Стоит отметить, что алгоритм срабатывает более корректно на выборках, в которых не все точки разбиваются на отдельные кластеры.

Кластеризация белковых последовательностей

Вычислительный эксперимент проводился на записях из базы данных UniProtKB, размер выборки – порядка 10^6. Входными параметрами алгоритма были $k = 0.3$ и число точек $\rho-$сети $n = 15$. Записи нарезались на слова длины 7. Поскольку количество кластеров для данной задачи априори неизвестно, алгоритм останавливался в случае резкого ухудшения качества. Для сравнения результатов применялся алгоритм *k*-means, который в качестве входного параметра получал число кластеров, найденное алгоритмом ранговой кластеризации на предыдущем шаге. Результаты работы показаны в табл. 2.

Табл. 2. Сравнение результатов работы алгоритмов на последовательностях аминокислотных остатков

Алгоритм	Число кластеризованных точек/ всего точек в выборке, M/N	Найдено кластеров	Качество кластеризации, Q	Сложность алгоритма
k-means	1	12	0.15	$O(N^{mK+1}\log N)$
ранговый	0.7	12	0.17	$O(nN\log N)$

Последняя колонка показывает сложности двух алгоритмов, здесь m – размерность пространства точки в алгоритме *k*-means, равна общему числу точек $m \equiv N$, K – число кластеров.

ЗАКЛЮЧЕНИЕ

Предлагаемому алгоритму кластеризации, в отличие алгоритмов кластеризации типа *k*-means, не требуется признаковое описание объектов, достаточно только матрицы парных расстояний. В связи с тем, что используются только ранговые значения набора расстояний от некоторой точки до всех остальных, предложенный алгоритм нечувствителен к «небольшим» изменениям функции расстояний, что важно, если у исследователя нет точной информации о виде этой функции. Предложенный алгоритм может быть использован для обнаружения мотивов в цепочках аминокислот, эта проблема сейчас активно обсуждается в биоинформатике [15,16].

СПИСОК ЛИТЕРАТУРЫ

1. Рудаков К.В., Торшин И.Ю. Об отборе информативных значений признаков на базе критериев разрешимости в задаче распознавания вторичной структуры белка. *Доклады Академии наук*. 2011. Т. 441. № 1. С. 24–28.
2. Рудаков К.В., Торшин И.Ю. Анализ информативности мотивов на основе критерия разрешимости в задаче распознавания структуры белка. *Информатика и её применения.* 2012. Т. 6. № 1.
3. *About Nucleotide And Protein Sequence Formats*. URL: http://www.ebi.ac.uk/help/formats.html (дата обращения: 17.05.2012).
4. *UniProtKB protein knowledgebase:* example of a record. URL: http://www.uniprot.org/uniprot/Q08753 (дата обращения: 20.11.2011).
5. Huang Z.A. Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. *Cooperative Research Centre for Advanced Computational Systems*. 1997.
6. Cardot H., Cenac P., Monnez J.-M. Fast clustering of large datasets with sequential k-medians: a stochastic gradient approach. *arXiv*. 2011. 1101.4179. URL: http://arxiv.org/abs/1101.4179 (дата обращения: 16.05.2012).
7. Seber G.A.F. *Multivariate Observations*. Hoboken, NJ: John Wiley & Sons, Inc. 1984.
8. Spath H. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples*. Translated by J. Goldschmidt. New York: Halsted Press, 1985.
9. Wei C.-P., Lee Y.-H., Hsu C.-M. Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems with Application.* 2003. Т. 24. № 4.
10. Giannopoulos P., Knauer C., Wahlstrom M., Werner D. Hardness of discrepancy computation and epsilon-net verification in high dimension. *arXiv*. 2011. 1103.4503. URL: http://arxiv.org/abs/1103.4503 (дата обращения: 16.05.2012).
11. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов. *Доклады Академии наук СССР*. 1965. Т. 163. № 4. С. 845–848.
12. *UniRef DataBase*. URL: http://www.uniprot.org/uniref/ (дата обращения: 13.05.2012).
13. *Protein knowledgebase UniprotKB*. URL: http://www.uniprot.org (дата обращения: 13.05.2012).
14. Kabsch W., Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers.* 1983. V. 22. № 12. P. 2577–637.
15. Marschall T. *Algorithms and Statistical Methods for Exact Motif Discovery*: PhD Thesis. Dortmund, 2011. URL: https://eldorado.tu-dortmund.de/bitstream/2003/27760/1/dissertation.pdf (дата обращения: 16.05.2012).

16. Li G., Chan T.M., Leung K.S., Lee K.H. A Cluster Refinement Algorithm for Motif Discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2010. V. 7. № 4. P. 654–668.

Материал поступил в редакцию 25.04.2012, опубликован 25.06.2012.