На правах рукописи

Фурлетова Евгения Игоревна

Оценка достоверности кластеров функционально-значимых фрагментов биологических последовательностей

03.01.09 Математическая биология, биоинформатика

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата физико-математических наук

Москва – 2012

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте математических проблем биологии РАН

Научный руководитель: доктор физико-математических наук
Михаил Абрамович Ройтберг

Официальные оппоненты: Всеволод Юрьевич Макеев
доктор физико-математических наук,
Федеральное государственное бюджетное учреждение науки
Институт общей генетики
им. Н.И. Вавилова РАН,
заведующий лабораторией

Иван Владимирович Кулаковский
кандидат физико-математических наук,
Федеральное государственное бюджетное учреждение науки
Институт молекулярной биологии
им. В.А. Энгельгардта РАН,
научный сотрудник

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт теоретической и экспериментальной биофизики Российской академии наук

Защита состоится «_____» _____2012 г. в _____ч. на заседании диссертационного совета Д 002.077.04 на базе Федерального государственного бюджетного учреждения науки Института проблем передачи информации РАН им. А.А. Харкевича по адресу 127994, Москва, ГСП-4, Большой Каретный переулок, д.19, стр.1.

С диссертацией можно ознакомиться в библиотеке Федерального государственного бюджетного учреждения науки Института проблем передачи информации им. А.А. Харкевича РАН.

Автореферат разослан «_____»_____2012 г.

Ученый секретарь диссертационного совета Д 002.077.04
доктор биологических наук, профессор Г.И. Рожкова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Одним из важных направлений биоинформатики является распознавание функционально-значимых участков биологических последовательностей. Как правило, такие участки отличаются повышенным содержанием определенных слов. Набор слов, характерных для класса функционально-значимых участков называется *мотивом*; место положения слова из мотива в биологической последовательности называется *вхождением* (или *сайтом* вхождения) мотива; набор сайтов, расположенных на анализируемом участке биологической последовательности, называется *кластером вхождений* (или *кластером сайтов*) мотива. Таким образом, задачу распознавания функционально-значимых участков можно решать путем поиска участков с повышенным содержанием слов из соответствующего мотива.

При таком подходе для решения задачи распознавания необходимо:

1) уметь оценивать достоверность (функциональную значимость) найденного кластера вхождений мотива и 2) определять характерную для мотива последовательность мономеров.

В биологических последовательностях кластеры вхождений определенного мотива могут появляться не только в результате естественного отбора, но и случайно. В качестве меры достоверности кластера вхождений мотива в биологической последовательности часто используется вероятность (*P-значение*) появления такого кластера в случайной последовательности соответствующей длины при подходящей вероятностной модели. Чем меньше вероятность обнаружить в случайной последовательности кластер, в котором количество вхождений мотива превышает заданный порог, тем больше оснований считать, что наличие такого кластера обусловлено его биологической значимостью.

Говоря более формально, *P-значением* кластера из r_0 вхождений мотива, расположенного на участке длины n_0 относительно заданной вероятностной модели, называется вероятность обнаружить в случайной последовательности длины n_0 кластер, содержащий не менее r_0 вхождений мотива. Таким образом, встает вопрос о разработке эффективного метода вычисления *P*-значения, что и является основной задачей данного исследования.

Проблема оценки достоверности предсказанных функционально-значимых вхождений мотива возникает в различных областях биоинформатики. Среди ученых, внесших значительный вклад в решение этой проблемы, О. Берг, Д. Гильберт, В. Ю. Макеев, А. А. Миронов, Г. Нюэль, М. Ренье, П. фон Хиппель и ряд других. Примером области, где необходима оценка достоверности кластеров сайтов, является идентификация потенциальных кластеров сайтов связывания белков – факторов регуляции транскрипции (ССФРТ), т. е. участков ДНК, которые взаимодействуют с факторами транскрипции. Здесь мотив – это набор до-

пустимых сайтов связывания данного фактора транскрипции. Во многих существующих программах распознавания кластеров ССФРТ отбор значимых результатов производится на основе вычисленного *P*-значения. Сказанное выше определяет актуальность выбранной темы.

В представленной работе рассматриваются только мотивы, в которых все слова имеют одну и ту же длину, которая называется *длиной* мотива. Именно такими являются, например, мотивы, описывающие ССФРТ. *Размером* мотива называется количество слов в мотиве.

Цели и задачи исследования. Целью проведенного исследования является разработка эффективного метода точного вычисления вероятности (*P*-значения) обнаружения в случайной последовательности длины n_0 не менее r_0 вхождений заданного мотива для различных вероятностных моделей генерирования последовательностей.

Исходя из поставленной цели, были сформулированы следующие задачи.

1. Исследовать комбинаторные свойства графов перекрытий слов, в частности, – зависимость количества перекрытий слов, входящих в мотив от размера и длины мотива.

2. Разработать алгоритмы для нахождения точного *P*-значения кластера из r_0 вхождений мотива, расположенного на участке длины n_0, относительно трех видов вероятностных моделей: моделей Бернулли, Марковских моделей данного порядка и скрытых Марковских моделей (СММ).

3. Реализовать разработанные алгоритмы в виде компьютерных программ.

4. Исследовать целесообразность применения классификационных затравок для поиска локальных сходств в аминокислотных последовательностях, что в свою очередь позволяет строить мотивы, описывающие функционально-значимые участки последовательностей. Разработать методику построения классификационных алфавитов.

5. Применить разработанные программы для анализа реальных аминокислотных последовательностей.

Методы исследования. В работе использованы методы теории алгоритмов, теории вероятностей, математической статистики, теоретической молекулярной биологии и биоинформатики.

Достоверность результатов. Достоверность результатов обеспечивается адекватностью используемых методов теории алгоритмов, теории вероятностей, математической статистики и биоинформатики, верификацией математических моделей, а также сравнением аналитических результатов с результатами математического моделирования.

Научная новизна. Получена оценка для среднего количества перекрытий слов, входящих в случайный мотив при равномерном распределении вероятностей мотивов. Доказано, что среднее количество перекрытий линейно зависит от размера мотива и не зависит от его длины.

Для вычисления *P*-значений был разработан алгоритм SufPref в трех модификациях, соответствующих трем видам вероятностных моделей: моделям Бернулли, Марковским моделям произвольного порядка и скрытым Марковским моделям (СММ). В алгоритме SufPref впервые были использованы графы перекрытий слов, что обеспечило его преимущество в быстродействии по сравнению с ранее разработанными алгоритмами.

Личный вклад автора. Автору принадлежат все описанные в диссертации оригинальные теоремы, алгоритмы и программы.

Автору принадлежит разработка, анализ и реализация алгоритма SufPref для точного вычисления *P*-значения вхождений мотивов, см. [3], [8], а также проведение компьютерных экспериментов для сравнения SufPref с другими алгоритмами. Автором была доказана теорема о среднем числе перекрытий [7] (см. раздел 2.3) и ряд утверждений, приведенных в диссертации. В работе [2] автору принадлежит вычисление статистической значимости шаблонов неупорядоченных участков в белках. В работах, описанных в публикациях [1], [4], [5], [6], автору принадлежат алгоритмы для построения классификационных алфавитов. Автор реализовал разработанные алгоритмы в программе и построил соответствующие классификационные алфавиты и одиночные классификационные затравки над этими алфавитами.

Теоретическая и практическая значимость. Теоретическая значимость работы состоит в исследовании комбинаторных объектов, связанных с наборами слов, – графов перекрытий слов и затравок. Доказано, что среднее количество перекрытий слов в мотивах данного размера и данной длины при равномерном распределении вероятностей на множестве мотивов линейно зависит от размера мотивов и не зависит от их длины. Разработан алгоритм SufPref, который вычисляет вероятность обнаружения в случайной последовательности длины n_0 не менее r_0 вхождений заданного мотива. Распределение вероятностей для последовательностей длины n_0 может задаваться с помощью модели Бернулли, Марковской модели произвольного порядка или СММ. Разработана методика построения классификационных алфавитов.

Практическая значимость работы состоит в программной реализации разработанных алгоритмов. Программа и исходные коды доступны через интернет по адресу: http://server2.lpm.org.ru/bio/online/sf. Реализация предложенного метода используется при создании опытного образца программного комплекса «СИМВОЛ», разрабатываемого в рамках государственного контракта № 07.514.11.4004. Результаты исследования использовались при выполнении работ по темам «Сравнительный анализ структур белков и нуклеиновых кислот» (номер государственной регистрации 01.2.00409635), «Математические методы анализа белков и нуклеиновых кислот: связь между последовательностями, структурой и функцией» (номер государственной регистрации 01.2.00952309), а также

при выполнении проекта РФФИ 09-04-01053-а «Достоверность и полнота результатов при компьютерном анализе последовательностей биополимеров». Построенные классификационные алфавиты были использованы Л. Ноэ в программе YASS (http://bioinfo.lifl.fr/yass/). Эта программа успешно применяется для выравнивания аминокислотных последовательностей.

Апробация результатов. Материалы диссертации докладывались на международных и всероссийских конференциях, в том числе: на международной конференции по биоинформатике регуляции и структуры генома (BGRS, Новосибирск, 2008), на Московских международных конференциях по вычислительной молекулярной биологии (MCCMB, Москва, 2007, 2009, 2011), международной конференции "The 2nd International Conference BIRD-ALBIO" (Австрия, 2008).

Публикации. По материалам диссертации опубликовано 8 печатных работ общим объемом 100 страниц. Из них три статьи в реферируемых научных изданиях (общий объем – 76 страниц), в том числе две статьи – в изданиях, входящих в список ВАК (общий объем – 48 страниц), а также пять тезисов докладов и препринтов (общий объем – 24 страницы).

Основные положения, выносимые на защиту.

1. Разработанный алгоритм SufPref корректно вычисляет вероятность (P-значение) появления не менее r_0 вхождений мотива H, слова в котором имеют одинаковую длину m, в случайной последовательности длины n_0; распределение вероятностей на множестве случайных последовательностей может задаваться с помощью моделей Бернулли, Марковских моделей произвольного порядка и скрытых Марковских моделей.

2. Размер используемой памяти и время работы алгоритма SufPref описываются следующими формулами (ниже A – используемый алфавит, $OV(H)$ – множество перекрытий слов в мотиве H):

- для моделей Бернулли:

  память: $O(r_0 \times m \times |OV(H)| + m \times |H|)$;

  время: $O(n_0 \times r_0 \times (|OV(H)| + |H|))$;

- для Марковских моделей порядка K:

  память: $O(r_0 \times K \times |A|^{K+1} + r_0 \times m \times |OV(H)| + m \times |H|)$;

  время: $O(n_0 \times r_0 \times (K \times |A|^{K+1} + |OV(H)| + |H|))$;

- для СММ:

  память: $O(|Q|^2 \times (|OV(H)| + |H|) + |Q| \times r_0 \times m \times |OV(H)| + m \times |H|)$;

  время: $O(|Q|^2 \times n_0 \times r_0 \times (|OV(H)| + |H|))$.

Полученные оценки показали, что SufPref превосходит по характеристикам большинство существующих алгоритмов.

3. Среднее количество перекрытий слов в мотивах данного размера и данной длины при равномерном распределении вероятностей на множестве мотивов линейно зависит от размера мотивов и не зависит от их длины.

4. Разработанная методика построения классификационных алфави-

тов позволяет получать классификационные затравки, которые не уступают лучшим из ранее известных видов затравок по чувствительности и избирательности.

Структура и объем работы. Диссертация состоит из введения, пяти глав, заключения и списка литературы (130 наименований). Полный объем диссертации составляет 128 страниц, количество рисунков – 20, количество таблиц – 20.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Введение

Во введении дана общая характеристика работы.

Глава 1. Обзор литературы

Глава 1 посвящена обзору литературы, отмечена связь проанализированных работ с предметом исследования диссертации. В этой главе также приведены основные определения и обозначения.

Глава 2. Теоретическая основа алгоритма SufPref

2.1. Постановка задачи

Пусть даны: алфавит $A = \{a_1, \ldots, a_{|A|}\}$; мотив $H = \{h_1, \ldots, h_s\}$, слова в котором имеют одинаковую длину m; вероятностное распределение на множестве слов заданной длины в алфавите A.

Целью работы является вычисление вероятности (*P-значения*) встретить мотив H по крайней мере r_0 раз в случайной последовательности длины n_0. Будем предполагать, что $n_0 \geq m$ и $r_0 \geq 1$. В данной работе рассматриваются три вида вероятностных моделей: модели Бернулли, Марковские модели порядка K, где $K \leq m$, и скрытые Марковские модели.

2.2. Перекрытия слов в мотиве. Граф перекрытий

В данном разделе вводятся понятия, относящиеся к структуре мотива. В частности, определяются левое и правое деревья перекрытий – основные структуры, используемые при работе алгоритма.

Определение 1. Слово w будем называть *перекрытием* в мотиве H, если существуют слова $h_1, h_2 \in H$ такие, что w является префиксом h_1 и суффиксом h_2. Множество всех перекрытий для H будем обозначать через $OV(H)$.

Определение 2. Пусть $w \in OV(H) \cup H$. Определим левого и правого предшественника $lpred(w)$ и $rpred(w)$ слова w следующим образом:

$$lpred(w) = \max\{u \in OV(H) \mid u \text{ является префиксом } w\};$$
$$rpred(w) = \max\{u \in OV(H) \mid u \text{ является суффиксом } w\}.$$

Определение 3. Два слова из мотива H называются эквивалентными, если их левые и правые предшественники совпадают. Множество всех классов эквивалентности для мотива H будем обозначать через H^*. Пусть $h^* \in H^*$ и $h \in h^*$. Тогда, по определению,

$$lpred(h^*) = lpred(h);\ rpred(h^*) = rpred(h).$$

Определение 4. *Левым деревом перекрытий* LOG_H называется дерево $LOG_H = <V(H), E_{LOG}(H)>$, где $V(H) = OV(H) \cup H^*$ и

$$\forall s, t \in V(H): (s, t) \in E_{LOG}(H) \Leftrightarrow s = lpred(t).$$

Аналогично, *правым деревом перекрытий* ROG_H называется дерево $ROG_H = <V(H), E_{ROG}(H)>$, где

$$\forall s, t \in V(H): (s, t) \in E_{ROG}(H) \Leftrightarrow s = rpred(t).$$

Графом перекрытий OVG_H будем называть объединение левого и правого деревьев перекрытий.

Для Марковских моделей определения деревьев перекрытий отличаются от приведенных выше техническими деталями (см. раздел 2.2 диссертации). В автореферате эти детали не рассматриваются.

2.3. Среднее число перекрытий

Время работы алгоритма SufPref пропорционально количеству перекрытий слов в мотиве. Поэтому была исследована зависимость числа перекрытий слов в мотиве от количества слов s и длины слов m в этом мотиве. Были рассмотрены две постановки задачи.

Во-первых, было исследовано среднее количество перекрытий для случая, когда все мотивы, размера s (то есть, содержащие s слов) и длины m равновероятны.

Теорема 1. Пусть $|A| \geq 2$ и $m \geq 2$. Тогда среднее количество $overlap\_avg(s, m)$ перекрытий в мотивах размера m и длины s не превосходит $c \cdot s$, где

$$c = \frac{|A|+1}{|A|-1} \cdot \sqrt{\frac{|A|^m}{|A|^m - 1}}.$$

Отметим, что если $|A| = 2$, то $c \approx 3{,}47$; если $|A| = 4$, то $c \approx 1{,}73$ и если $|A| = 20$, то $c \approx 1{,}11$.

Гипотеза о линейной зависимости среднего числа перекрытий слов $overlap\_avg(s, m)$ от размера мотива s и отсутствии зависимости этого числа от длины слов m в мотиве была с помощью компьютерных экспериментов проверена для случая, когда вероятности отдельных слов в мотиве описываются неравномерным распределением Бернулли, а вероятность мотива – это произведение вероятностей входящих в него слов. В экспериментах рассматривался 4-буквенный алфавит, проверялись значения m от 8 до 50 с шагом 4 и значения s от 100 до 1000 с шагом 100, а также два распределения Бернулли – равномерное и неравномерное (с вероятностями $\{0{,}1; 0{,}2; 0{,}3; 0{,}4\}$). Эксперименты показали, что в обоих случаях $overlap\_avg(s, m) \approx s$. Результаты экспериментов для неравномерного распределения представлены на рисунках 1 и 2, для равномерного распределения результаты аналогичны.

Во-вторых, было подсчитано количество перекрытий для мотивов, которые задаются с помощью двух матриц позиционных весов (МПВ), описывающих регуляторные сайты *Drosophila*, и различных порогов. Длины

слов в мотивах, заданных с помощью этих матриц, равны соответственно $m = 8$ и $m = 12$. Значения порогов выбиралось так, чтобы размеры мотивов были примерно от 100 до 1000 слов. Показано, что отношение количества перекрытий к количеству слов в мотивах меняется от 0,04 до 0,09 для матрицы длины 8 и от 0,06 до 0,12 для матрицы длины 12.

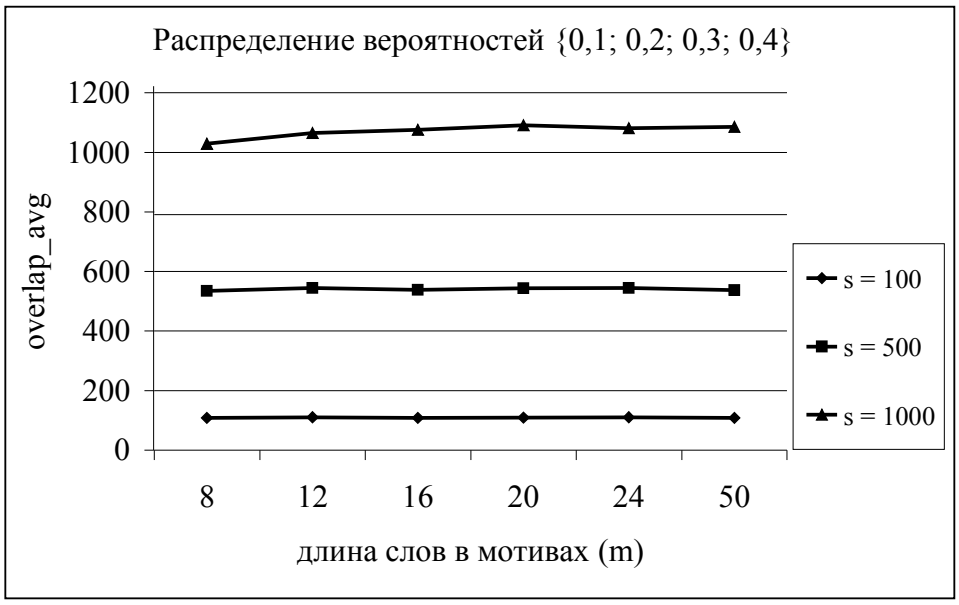

Рис. 1. Зависимость среднего числа перекрытий *overlap_avg*(*m*,*s*) от параметров *m* и *s*, где *m* = 8, 12, 16, 20, 24 и *s* = 100, 500, 1000.

Рис. 2. Зависимость среднего числа перекрытий *overlap_avg*(*s*,*m*) от количества слов s в наборах при данной длине слов *m* = 10, где $s = 100 \cdot i$, $i = 1,\ldots,10$.

2.4. Текстовые множества

Для краткости символьные последовательности, в которых рассматриваются вхождения мотивов ниже называются *текстами*.

Пусть даны натуральные числа *n* и *r*. Определим множество:

$$B(n, r) = \{T \in A^n | \ T \text{ содержит не менее } r \text{ вхождений мотива } H\}.$$

Очевидно, *P-значение* кластера из r_0 вхождений мотива H, расположенного на участке длины n_0, равно вероятности $Prob(B(n_0, r_0))$. Приведем ряд определений и обозначений, которые используются в диссертации и связаны с перекрытиями слов.

Пусть $x,w \in OV(H) \cup H$, x – префикс w; $h^* \in H^*$.Тогда:

- $len(x)$ – длина x;
- $H(x)$ – подмножество слов из H, заканчивающихся на x;
- $OverlapPrefix(w) = \{x \in OV(H) |\ x$ является префиксом $w\}$.

Если x – префикс w, то $Back(x, w)$ – такой суффикс слова w, что $w = x \cdot Back(x, w)$. По определению:

$$Back(w) = Back(lpred(w), w);\ Back(h^*) = \bigcup_{h \in h^*} Back(h).$$

Алгоритм SufPref для вычисления вероятностей множеств $B(n, r)$ использует ряд текстовых множеств, которые определены ниже. Пусть $h \in H$ и $w \in OV(H)$. Тогда

$$R(n,r,h) = \{T \in A^n |\ T \text{ заканчивается на } h \,\&\, T \text{ содержит ровно } r \text{ вхождений } H\};$$

$$E(n,r,h) = \{T \in A^n |\ T \text{ заканчивается на } h \,\&\, T \text{ содержит не менее } r \text{ вхождений } H\};$$

$$D(n,r,w) = \bigcup_{x \in OverlapPrefix(w)} R(n - len(Back(x,w)), r, H(x)) \cdot Back(x,w);\ (w \neq \varepsilon)$$

$$D(n,r,\varepsilon) = \varnothing.$$

Соответственно, для класса эквивалентности $h^* \in H^*$

$$R(n,r,h^*) = \sum_{h \in h^*} R(n,r,h);\ E(n,r,h^*) = \sum_{h \in h^*} E(n,r,h).$$

Для указанных текстовых множеств доказаны следующие соотношения.

1. $B(n,r) = B(n-1,r) \cdot A \cup R(n,r,H)$ (1)
2. $R(n,r,h^*) = E(n,r,h^*) \setminus E(n,r+1,h^*)$ (2)
3. Если $r > 1$, то
   $E(n,r,h^*) = B(n-m,r-1) \cdot h^* \cup D(n - len(Back(h^*)), r, lpred(h^*)) \cdot Back(h^*)$ (3)
   если $r = 1$, то
   $E(n,1,h^*) = A^{n-m} \cdot h^*$ (4)
4. $D(n,r,w) = D(n - len(Back(w)), r, lpred(w)) \cdot Back(w) \cup R(n,r,H(w)),\ w \neq \varepsilon$ (5)
5. $R(n,r,H(w)) = \left(\bigcup_{\substack{x \in OV(H), \\ w = rpred(x)}} R(n,r,H(x)) \right) \cup \left(\bigcup_{\substack{h \in H^*, \\ w = rpred(h^*)}} R(n,r,h^*) \right).$ (6)

2.5. Вероятности текстовых множеств

В разделе 2.5 приведены формулы вычисления вероятностей описанных выше множеств для различных вероятностных моделей. Вывод формул непосредственно следует из свойств используемых моделей и того, что все объединения в формулах (1)–(6) – дизъюнктные. Например, для

множества $B(n, r)$ и модели Бернулли получаем следующую формулу

$$Prob(B(n,r))=Prob(B(n-1,r))\cdot Prob(A)+Prob(R(n,r,H))=\sum_{k=m}^{n} Prob(R(k,r,H)).$$

Глава 3. Алгоритм SufPref для вычисления *P*-значения участка, содержащего кластер вхождений мотива

Разделы 3.1–3.3. Общее описание алгоритма SufPref

В разделах 3.1– 3.3 дано описание вариантов алгоритма SufPref, ориентированных соответственно на вероятностные модели Бернулли, СММ и Марковские модели. Различия между вариантами носят технический характер. Для краткости ниже описана только общая основа всех вариантов алгоритма SufPref. Во всех случаях основными структурами данных являются деревья перекрытий LOG_{H} и ROG_{H}.

Целью алгоритма SufPref является нахождение вероятности $Prob(B(n_0, r_0))$. Алгоритм SufPref в цикле по $n = m + 1, ..., n_0$ и, для каждого значения n, в цикле по $r = 1, ..., r_0$ вычисляет следующие вероятности:

- $Prob(B(n - m, r))$, если $n > 2m$;
- $Prob(R(n, r, h^*))$ для всех $h^* \in H^*$;
- $Prob(R(n, r, H(w)))$ для всех $w \in OV(H)$.

Для вычисления $Prob(B(n - m, r))$ используется соотношение (1). Вычисления $Prob(R(n, r, h^*))$ выполняются путем обхода левого дерева перекрытий LOG_H от корня к листьям. Сначала каждой из внутренних вершин $w \in OV(H)$ вычисляются вероятности $Prob(D(n - |Back(w)|, r, w))$ согласно (5). При этом используются значения аналогичных вероятностей, вычисленные для ранее просмотренных вершин. Далее, для листьев $h^* \in H^*$ дерева LOG_H алгоритм вычисляет $Prob(E(n, r, h^*))$ $(r = 1, ..., r_0 + 1)$, используя вероятности $Prob(D(n - |Back(w)|, r, w))$ и формулы (3) и (4); а затем – $Prob(R(n, r, h^*))$ согласно (2). Наконец, используя обход ROG_H от листьев к корню алгоритм вычисляет вероятности $Prob(R(n, r, H(w)))$ для всех $w \in OV(H)$ согласно (6).

Отметим, что $Prob(R(n, r, H(\varepsilon))) = Prob(R(n, r, H))$.

На заключительном этапе алгоритм SufPref находит вероятности $Prob(B(n_0 - m+1, r_0))$, …, $Prob(B(n_0, r_0))$, используя формулу (1) и ранее вычисленные вероятности.

3.4. Предварительные построения

В этом разделе описаны алгоритмы построения графов перекрытий и инициализации структур данных, выполняемые на предварительном этапе работы SufPref.

3.5. Анализ сложности алгоритма SufPref

В этом разделе дан теоретический анализ сложности алгоритма SufPref.

Теорема 2. Время работы алгоритма составляет:

- $O(n_0 \times r_0 \times (|OV(H)|+|H|))$ для вероятностой модели Бернулли;
- $O(|Q|^2 \times n_0 \times r_0 \times (|OV(H)|+|H|))$ – для СММ;
- $O(n_0 \times r_0 \times (K \times |A|^{K+1}+|OV(H)|+|H|))$ – для Марковской модели порядка K.

Размер используемой алгоритмом памяти составляет

- $O(r_0 \times m \times |OV(H)|+m \times |H|)$ для вероятностной модели Бернулли;
- $O(|Q|^2 \times (|OV(H)|+|H|)+|Q| \times r_0 \times m \times |OV(H)| +m \times |H|)$ – для СММ;
- $O(r_0 \times K \times |A|^{K+1}+r_0 \times m \times |OV(H)|+m \times |H|)$ – для Марковской модели порядка K.

Глава 4. Программная реализация алгоритма SufPref и ее апробация

4.1. Программная реализация алгоритма SufPref

Алгоритм SufPref для точного вычисления P-значения был реализован в программе на языке C++, работающей под Unix и Windows. Программа и подробная документация к ней доступны на сайте http://server2.lpm.org.ru/bio/online/sf. Созданы следующие версии программы: 1) Веб-версия; 2) версия, запускаемая с командной строки; 3) реализация в виде Python-модуля.

Входными параметрами программы являются: 1) алфавит A; 2) вероятностное распределение случайных последовательностей; 3) длина n_0 случайной последовательности; 4) мотив H; 5) минимальное количество r_0 вхождений мотива. На выходе программа выдает найденное P-значение. Все версии программы поддерживают работу с вероятностными моделями Бернулли, Марковскими моделями произвольных порядков и СММ.

4.2. Сравнение программы SufPref с программой AhoPro

В разделе 4.2. приведены результаты сравнения программы SufPref с программой AhoPro[1], поддерживающей работу с моделями Бернулли и Марковскими моделями 1-го порядка. Программа AhoPro была выбрана для сравнения по двум причинам: во-первых, соответствующий ей алгоритм AhoPro имеет лучшие оценки времени работы и размера используемой памяти среди подобных алгоритмов; во-вторых данная программа эффективно применяется при решении задач биоинформатики. Для сравнения программ было вычислено P-значение для следующих наборов тестовых данных:

1) алфавит – {A, C, G, T};

2) вероятностное распределение на последовательностях задано следующими моделями:

- моделью Бернулли с вероятностями букв алфавита {0,3; 0,3; 0,2; 0,2};

[1] Boeva V., et al. Exact p-value calculation for heterotypic clusters of regulatory motifs and its application in computational annotation of cisregulatory modules. // Algorithms for molecular biology. 2007. Vol. 2, N. 13. P. 25. Программа AhoPro доступна по адресу http://favorov.imb.ac.ru/ahokocc/.

- Марковской моделью первого порядка, условные вероятности которой заданы матрицей

$$\begin{pmatrix} 0{,}3 & 0{,}3 & 0{,}2 & 0{,}2 \\ 0{,}3 & 0{,}3 & 0{,}2 & 0{,}2 \\ 0{,}3 & 0{,}3 & 0{,}2 & 0{,}2 \\ 0{,}3 & 0{,}3 & 0{,}2 & 0{,}2 \end{pmatrix},$$

3) длина случайной последовательности $n_0 = 1000$;

4) минимальное количество вхождений $r_0 = 10$;

5) мотивы двух видов: (а) 10 и 11 случайных мотивов длины 8 и 12 соответственно (буквы имеют равномерное распределение); (б) 9 и 11 мотивов длины 8 и 12 соответственно, заданных матрицами позиционных весов и различными порогами.

Результаты сравнения показали: 1) для модели Бернулли SufPref работает в 4–20 раз быстрее AhoPro и использует в среднем в полтора раза меньше памяти; 2) для Марковской модели SufPref работает в 2–5 раз быстрее AhoPro, но проигрывает по размеру используемой памяти не более чем на 20%.

4.3. Оценка статистической значимости шаблонов неупорядоченных участков в белках

Программа SufPref была использована при нахождении статистической значимости шаблонов неупорядоченных участков в белках, то есть участков, пространственная структура которых не поддается определению методами кристаллографии. О. В. Галзитской и М. Ю. Лобановым была построена библиотека из 109 шаблонов, которые описывают неупорядоченные участки. Каждый шаблон – это слово в аминокислотном алфавите длиной от 6 до 50. Мотив, соответствующий шаблону, содержит все слова той же длины, которые отличаются от шаблона не более, чем в 20% позиций и удовлетворяют некоторым дополнительным условиям. Для каждого такого мотива h было подсчитано количество $k(h)$ последовательностей белков, содержащих этот мотив, а также соответствующее Z-значение

$$Z(h,n) = \frac{k(h) - N \cdot P(h,n)}{\sqrt{N \cdot P(h,n) \cdot (1 - P(h,n))}},$$

где $P(h, n)$ – вероятность (P-значение) хотя бы один раз обнаружить h в случайной последовательности длины n (n полагалось равным 260 – средней длине белка в базе данных); $k(h)$ – число последовательностей в базе данных, содержащих не менее одного вхождения h; $N = 28727$ – количество уникальных (не гомологичных друг другу по аминокислотной последовательности) белков в базе данных.

Для шаблонов, длина которых равна 15 и меньше, для вычисления $P(h, n)$ использовалась разработанная программа SufPref. Отметим, что количество слов в множестве H экспоненциально растет с увеличением длины шаблона h, поэтому точное вычисление P-значения для длинных

шаблонов становится невозможным. Поэтому для шаблонов, длина которых равна 16 и больше, использовалась оценка данной вероятности сверху $G(h, n) \geq P(h, n)$, где $G(h, n) = (n - m + 1) \cdot Prob(H)$.

Предполагалось, что шаблон h является статистически значимым, если $Z(h, n)$ больше, чем определенный q-квантиль. Были рассмотрены 99-квантиль и 95-квантиль, значения которых равны 2,33 и 1,65 соответственно. Оказалось, что для 102 и 106 из 109 шаблонов Z-значение больше, чем 2,33 и 1,65 соответственно. Также оказалось, что 89 из 109 шаблонов имеют Z-значение больше 5, то есть являются перепредставленными в белках.

Глава 5. Построение классификационных затравок для нахождения локальных сходств аминокислотных последовательностей

Глава 5 посвящена поиску локальных сходств - одному из важных методов построения мотивов. Обнаружив ряд сходных между собой фрагментов в одном геноме или в нескольких геномах, можно по ним построить мотив, который затем будет использоваться для поиска участков генома, содержащих избыточное количество вхождений таких мотивов.

Для решения задачи поиска локальных сходств, как правило, применяются фильтрационные методы: сначала находят короткие гомологичные фрагменты последовательностей (якоря или затравочные сходства). Затем якоря расширяют до выравниваний более длинных фрагментов путем анализа окрестностей якорей. Образец для поиска затравочных сходств называется *затравкой*. Фильтрационные методы работают достаточно быстро, однако ценой за это может быть понижение чувствительности (доли найденных сходств). Поэтому встает вопрос о таком выборе затравки, чтобы алгоритм работал быстро и при этом имел достаточно высокую чувствительность. Для нуклеотидных последовательностей в этом направлении были получены хорошие результаты. Для аминокислотных последовательностей единственным удовлетворительным решением этой проблемы стало использование векторных затравок. В данной главе исследована возможность использования для анализа аминокислотных последовательностей классификационных затравок, ранее использовавшихся только при работе с нуклеотидными последовательностями.

5.1. Классификационные затравки. Основные определения

Пусть дан алфавит аминокислот A. Обозначим через $\Pi = \{<x, y> | \, x, y \in A\}$ множество пар аминокислот.

Определение 5. Алфавит B называется *классификационным*, если:

- каждой букве α алфавита B соответствуют подмножество $\Pi(\alpha)$ множества Π, разным буквам соответствуют разные подмножества;
- B содержит букву # такую, что $\Pi(\#) = \{<x, x> | x \in A\}$, т.е. # означает совпадение символов, сопоставленных при выравнивании;
- множества, соответствующие буквам из алфавита B симметричны,

т. е. для любых аминокислот $x, y \in A$ и буквы $\alpha \in B$ выполняется:

$$\langle x, y\rangle \in \Pi(\alpha) \Leftrightarrow \langle y, x\rangle \in \Pi(\alpha).$$

Классификационной затравкой (одиночной) называется слово в алфавите B. Пусть $s_1, s_2 \in A^n$. Будем говорить, что выравнивание (s_1, s_2) образует *затравочное сходство* относительно $\pi = \alpha_1 \ldots \alpha_n$, если $\langle s_1[i], s_2[i]\rangle \in \Pi(\alpha_i)$ для всех $i = 1, ..., n$. Будем говорить, что локальное сходство *допускается* затравкой π, если оно содержит затравочное сходство относительно π.

Множественной затравкой называется группа одиночных затравок. Локальное сходство допускается множественной затравкой, если оно допускается какой либо входящей в нее одиночной затравкой.

Затравка может быть охарактеризована двумя параметрами: *чувствительностью* и *избирательностью*. Говоря неформально, чувствительность (относительно заданного множества целевых сходств) – вероятность того, что случайное выравнивание из заданного множества целевых выравниваний допускается затравкой; избирательность – вероятность в рамках выбранной вероятностной модели того, что независимые случайные последовательности не образуют затравочное сходство. Приведем формальные определения.

Пусть для каждой аминокислоты $x \in A$ задана вероятность $b(x)$ ее появления в аминокислотных последовательностях. *Базовой вероятностью* $b(x, y)$ пары аминокислот $\langle x, y\rangle \in \Pi$ называется вероятность $b(x, y) = b(x) \cdot b(y)$. Кроме того, для каждой пары аминокислот $\langle x, y\rangle \in \Pi$ задана *целевая вероятность* $f(x, y)$. Говоря неформально, $f(x, y)$ – это вероятность встретить сопоставление (x, y) в выравнивании сходных участков. *Целевой вероятностью выравнивания* слов $s_1, s_2 \in A^n$ называется произведение

$$f(s_1, s_2) = \prod_{k=1}^{n} f(s_1[k], s_2[k]).$$

Для классификационной буквы $\alpha \in B$ и классификационной затравки $\pi = \alpha_1 \ldots \alpha_n$ определим:

базовые вероятности:

$$b(\alpha) = \sum_{\langle x, y\rangle \in \Pi(\alpha)} b(x) \cdot b(y);\ b(\pi) = \prod_{k=1}^{n} b(\alpha_k);$$

целевые вероятности:

$$f(\alpha) = \sum_{\langle x, y\rangle \in \Pi(\alpha)} f(x, y);\ f(\pi) = \prod_{k=1}^{n} f(\alpha_k).$$

Избирательностью затравки π называется величина $1 - b(\pi)$. *Чувствительностью* затравки π относительно заданного множества целевых сходств называется сумма целевых вероятностей тех сходств, которые допускаются затравкой π.

5.2. Классификационные алфавиты

В диссертации предложены три вида классификационных алфавитов: пороговый алфавит, иерархический транзитивный алфавит (ИТА) и неиерархический транзитивный алфавит (НТА).

Пороговый алфавит

Для каждой пары аминокислот $p \in \Pi$ рассмотрим отношение

$$h(p) = f(p)/b(p).$$

Упорядочим все пары аминокислот по возрастанию $h(p)$. Пусть дан набор порогов $T = \{t_1, \ldots, t_s\}$. Каждому порогу t сопоставим классификационную букву $\alpha(t)$, которой соответствует множество пар аминокислот

$$\Pi(\alpha(t)) = \{p \in \Pi | f(p)/b(p) \geq t\}.$$

Такие буквы будем называть *пороговыми. Пороговым алфавитом* будем называть совокупность букв

$$B(T) = \{\alpha(t) \mid t \in T\}.$$

Пороговые буквы являются вложенными, то есть для любых двух порогов $t_1, t_2 \in T$, где $t_1 > t_2$, выполняется:

$$\Pi(\alpha(t_1)) \subseteq \Pi(\alpha(t_2)).$$

Транзитивные алфавиты

Классификационную букву α будем называть *транзитивной*, если и только если

$$\langle x, y\rangle, \langle y, z\rangle \in \Pi(\alpha) \Rightarrow \langle x, z\rangle \in \Pi(\alpha).$$

Соответственно, классификационную затравку будем называть *транзитивной*, если она составлена из транзитивных букв. Транзитивная затравка π задает отношение эквивалентности на аминокислотных последовательностях той же длины. Мотивацией к построению транзитивных алфавитов является уменьшение количества списков хэш-таблицы, к которым приходится обращаться при поиске локальных сходств.

В работе рассматриваются два вида транзитивных алфавитов: иерархический и неиерархический. Иерархическим называется алфавит, в котором буквы вложены друг в друга. Соответственно, в неиерархическом алфавите буквы не обязательно являются вложенными.

5.3. Классификационные затравки

Эффективность построенных алфавитов была продемонстрирована путем построения на их основе затравок, вычисления чувствительности и избирательности этих затравок и сравнения характеристик построенных затравок с аналогичными характеристиками затравок других типов.

Автором диссертации с помощью программы Iedera[2] были построены одиночные классификационные затравки длины 3–5, избирательности которых принадлежат промежутку [0,988; 1]. Также с помощью Iedera были вычислены чувствительности этих затравок, и среди затравок было отобрано Парето-множество, т. е. множество, в котором не су-

[2] Kucherov G., et. al. A unifying framework for seed sensitivity and its application to subset seeds. // Journal of Bioinformatics and Computational Biology. 2006. Vol. 4, N. 2. P. 553–570.

ществует затравки, которая одновременно имеет меньшую чувствительность и избирательность, чем некоторая другая затравка из этого множества. При вычисления чувствительности в качестве целевых выравниваний рассматривались выравнивания аминокислотных последовательностей длины 16 и 32. Целевое распределение вероятностей соответствовало матрице замен BLOSUM62.

Построенные одиночные классификационные затравки сравнивались с векторной затравкой, которая используется в программе BLASTp. Затравочными сходствами относительно затравки BLASTp являются выравнивания длины три, суммарный вес пар аминокислот в позициях которых (относительно заданной весовой матрицы) больше некоторого порога. Результаты показали, что чувствительности и избирательности затравки BLASTp и одиночных классификационных затравок примерно равны. Также было показано, что пороговые затравки немного превосходят транзитивные.

Используя полученные в диссертационной работе классификационные алфавиты, Л. Ноэ в работе[3] построил множественные классификационные затравки. Построенные затравки сравнивались с затравкой BLASTp с порогами 10–13 и множественной векторной затравкой[4] с порогами 13–16. При сравнении были получены следующие результаты:

- Множественные пороговые затравки сравнимы по чувствительности и избирательности с множественной векторной затравкой и превосходят транзитивные классификационные затравки, а также затравку BLASTp.
- Множественные пороговые затравки превосходят затравку BLASTp по чувствительности примерно на 5% при примерно равных избирательностях. Например при избирательности 0,999355 (соответствует затравке BLASTp с порогом 13) и при длине целевых выравниваний 32 чувствительность затравки BLASTp составляет 0,81925, а чувствительность лучшей из пороговых затравок – 0,869064.
- Множественные транзитивные затравки сравнимы с затравкой BLASTp при относительно невысоких значениях избирательности (ниже 0,997) и немного превосходят ее при более высоких значениях избирательности. При этом преимущество транзитивных затравок растет с ростом избирательности и длины целевых выравниваний.

Заключение

В диссертации представлены следующие основные результаты.

1. Разработан алгоритм SufPref для вычисления вероятности (P-значения) появления не менее r_0 вхождений заданного мотива в случайной последовательности длины n_0. Распределение вероятностей на множе-

[3] Noe L., et. al. On subset seeds for protein alignment. // IEEE/ACM Trans. Comput. Biol. Bioinformatics. 2009. Vol. 6, N. 3. P. 483–494.

[4] Brejová B., et. al. Vector seeds: An extension to spaced seeds. // Journal of Computer and System Sciences. 2005. Vol. 70, N. 3. P. 364–380; Brown D. Optimizing multiple seed for protein homology search. // IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2005. Vol. 2, N. 1. P. 29–38.

стве случайных последовательностей может задаваться с помощью моделей Бернулли, Марковских моделей порядка K, где $K \leq m$, и скрытых Марковских моделей.

2. Получены оценки сложности работы алгоритма для указанных видов вероятностных моделей.

3. С помощью компьютерных экспериментов было показано, что среднее число перекрытий $overlap\_avg(s, m)$ в случайных мотивах, слова в которых порождены в рамках моделей Бернулли, пропорционально количеству слов s в мотивах и не зависит от длины m этих слов. Для мотивов, слова в которых имеют равномерное распределение, была доказана теорема о том, что $overlap\_avg(s, m) \leq c \cdot s$, где c – константа, зависящая от размера алфавита; $c < 4$. Из этого следует, что скорость работы алгоритма в среднем не зависит от длины слов в мотиве.

4. Алгоритм SufPref реализован в виде кросс-платформенного программного комплекса.

5. Проведено сравнение реализации алгоритма SufPref с реализацией алгоритма AhoPro для модели Бернулли и Марковской модели первого порядка. Для модели Бернулли SufPref работает в 4–20 раз быстрее AhoPro и использует в среднем в полтора раза меньше памяти. Для Марковской модели SufPref работает в 2–5 раз быстрее AhoPro, но проигрывает по размеру используемой памяти не более чем на 20%.

6. Вычислены P-значения и статистические значимости вхождений мотивов неупорядоченных участков в белках.

7. Построены классификационные алфавиты для анализа аминокислотных последовательностей: (1) пороговый алфавит, (2) иерархический транзитивный алфавит; (3) неиерархический транзитивный алфавит. Затравки над этими алфавитами не уступают по чувствительности и избирательности лучшим из ранее известных видов затравок.

РАБОТЫ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

1. Roytberg M. A., Gambin A., Noe L., Lasota S., Furletova E. I., Szczurek E., Kucherov G. On subset seeds for protein alignment. // IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2009. Vol. 6, N. 3. P. 483–494.

2. Lobanov M. Y., Furletova E. I., Bogatyreva N. C., Roytberg M. A., Galzitskaya O. V. Library of disordered patterns in 3D protein structures. PLoS Computational Biology. 2010. Vol. 6, N. 10. P. 1–10.

3. Regnier M., Kirakosian Z., Furletova E. I., Roytberg M.A. A word counting gpaph. // London algorithmics 2008: theory and practice. 2009. P. 10–43.

4. Furletova E. I., Kucherov G., Noé L., Roytberg M. A., Tsitovich I. I. Statistical approach to the design of subset seeds for protein alignment. // Proceedings of the International Moscow Conference on computational molecular biology. July 2007. P. 94.

5. Furletova E. I., Kucherov G., Noé L., Roytberg M. A., Tsitovich I. I. Transitive subset seeds for protein alignment. // Proceedings of the 6th International Conference on Bioinformatics of Genome Regulation and Structure (BGRS). July 2008, Novosibirsk (Russia). P. 77.

6. Roytberg M. A., Gambin A., Noe L., Lasota S., Furletova E. I., Szczurek E., Kucherov G. Efficient seeding techniques for protein similarity search. // Proceedings of the 2nd International Conference BIRD-ALBIO. 2008. P. 466–478.

7. Regnier M., Furletova E. I., Roytberg M. A. An average number of suffix-prefixes. // Proceedings of the International Moscow Conference on computational molecular biology. 2009. P. 313–314.

8. Regnier M., Furletova E. I., Roytberg M. A., Yakovlev V. V. An algorithm for exact probability of pattern occurrences calculation. // Proceedings of the International Moscow Conference on computational molecular biology. 2011. P. 320.