Т. Ф. Валеев*

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ПОИСКА РЕГУЛЯТОРНЫХ МОДУЛЕЙ В ПОСЛЕДОВАТЕЛЬНОСТЯХ ДНК, ИСПОЛЬЗУЮЩИХ ДАННЫЕ МИКРОЭРРЭЕВ

ВВЕДЕНИЕ

В связи с появлением и развитием технологии микроэррэев для получения значений экспрессии (величины, соответствующей количеству белка, производимого геном) многих генов одновременно, встала задача определения набора транскрипционных факторов, которые регулируют экспрессию данной группы генов [1]. Говорят, что фактор регулирует экспрессию данного гена, если от наличия данного фактора в достаточной концентрации в клетке зависит величина экспрессии гена. Другими словами, задача состоит в том, чтобы найти набор транскрипционных факторов, которые связываются с промоторами (регуляторными участками) генов, показывающих высокую экспрессию.

Также используют данные из нескольких экспериментов по экспрессии для более точного предсказания. Тогда экспрессия гена представляется в виде вектора значений экспрессии в каждом эксперименте.

Задача поиска набора транскрипционных факторов, которые участвуют в регуляции экспрессии генов, решается в два этапа. Во-первых, ищутся потенциальные сайты связывания транскрипционных факторов с промоторами генов. Здесь на каждом промоторе по отдельности выполняется поиск сайтов связывания каждого фактора, вычислительная сложность пропорциональна числу промоторов и числу факторов. Эта часть, как правило, реализуется сходным образом в различных разработках, её обычно можно выполнить независимо, сохранив результаты для второго этапа.

Второй этап заключается в поиске композитного модуля, набора факторов, регулирующих данный набор генов. Здесь возможны значительные вариации в постановке задачи и методах её решения.

* lan@biorainbow.com

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 05-08-33448).

В настоящее время разработано несколько программных продуктов, которые позволяют решать задачу поиска регуляторных факторов. В данной статье рассмотрены три такие разработки. Это пакет TOUCAN [2], разработанный группой Bioi в Католическом Университете города Леувен, Бельгия; система TELiS [3], созданная в Калифорнийском университете, а также наша собственная разработка, Composite Module Analyst [4]. Ниже приведены описание и обзор возможностей каждой из этих систем, а затем проанализированы их достоинства и недостатки.

1. ПАКЕТ TOUCAN

Пакет TOUCAN представляет собой набор программ для решения различных задач в области анализа регуляторных последовательностей промоторов. Он включает в себя клиентское приложение на языке Java, которое загружает данные, решает часть задач на стороне пользователя, а также позволяет посредством технологии SOAP (Simple Object Access Protocol) связываться с веб-службами на Tomcat-сервере разработчиков и пользоваться процессорными ресурсами кластера на 10 процессоров для запуска особо трудоёмких вычислений. Управление распределёнными вычислениями осуществляется при помощи Java RMI (Remote Method Invocation). Пакет доступен на веб-сервере разработчиков:

`http://www.esat.kuleuven.ac.be/~saerts/software/toucan.php`

Среди доступных веб-служб были проанализированы MotifScanner и ModuleSearcher. Первое приложение используется для поиска сайтов связывания с транскрипционными факторами, а второе — для поиска транскрипционного модуля, регулирующего заданный набор генов.

На вход программы ModuleSearcher подаётся набор промоторов, показавших высокую экспрессию в эксперименте. Входные параметры включают количество факторов в модуле, максимально допустимое расстояние (количество нуклеотидов) между сайтами связывания отдельных факторов модуля на каждом промоторе. В основе поиска набора факторов лежит направленный перебор возможных наборов одним из двух способов, алгоритмом A* или генетическим алгоритмом. При этом максимизируется весовая функция, характеризующая, насколько хорошо данный набор факторов соответствует набору промоторов. Весовая функция учитывает, сколько факторов из пробного набора имеют сайты связывания на каждом из промоторов и насколько близко они расположены.

Алгоритм A* [5] представляет собой метод ветвей и границ, адаптированный под данную задачу. Метод ветвей и границ впервые был предложен Лендом и Дойгом [6] для решения общей задачи линейного программирования. Он позволяет найти оптимальное решение NP-полных задач, существенно снижая трудоёмкость вычислений в большинстве ситуаций. Тем не менее, в данной задаче вычисления могут занять много времени (до нескольких суток на современных процессорах) и изрядный объём памяти. Учитывая, что общее число факторов может превышать 100, число всевозможных модулей из пяти факторов превышает 10^{10}, практически невозможно искать оптимальный модуль, состоящий более чем из четырёх факторов. В большинстве случаев на практике имеет смысл использовать генетический алгоритм [7]. Этот алгоритм не находит гарантированно оптимального решения, но результаты тестирования показали, что модуль, найденный генетическим алгоритмом, в подавляющем большинстве случаев совпадает с оптимальным модулем, найденным алгоритмом A*. Генетический алгоритм выполняется на порядок быстрее и требует намного меньше памяти. Суть его состоит в следующем. Вначале генерируется популяция из случайных модулей (~200−1000), они сортируются в порядке убывания весов, после чего некоторая доля лучших (30−50%) остаётся (выживает) и дает «потомство», а остальные вымирают. Потомство создается уже не случайно, а посредством скрещивания существующих модулей (случайно выбираются два модуля и генерируется новый, содержащий часть факторов из первого, а часть — из второго) и мутации (в полученном после скрещивания модуле каждый фактор с некоторой вероятностью может быть заменён на случайный). Эта процедура повторяется заданное число итераций (поколений), порядка 100−1000 в зависимости от других входных параметров. Затем модуль в популяции, обладающий наибольшим весом, выдаётся как результат.

2. СИСТЕМА TELIS

Система TELiS также реализована на Java и разделена на два модуля, PromoterScan и PromoterStats, первый из которых выполняет поиск сайтов связывания, а второй находит факторы, которые наиболее (или наименее) представлены на заданной пользователем выборке промоторов. Чтобы не выполнять многократно поиск сайтов, авторы выполнили эту операцию для промоторов всех генов (~40000) человека и для всех известных регули-

рующих факторов позвоночных (~200). Результаты этого анализа сохранены в базе данных MySQL и доступны на сайте разработчиков:

`http://www.telis.ucla.edu/`

Также предоставляется возможность запустить через веб-интерфейс второй модуль PromoterStats, который выполняется на стороне сервера (Java-servlet) и выдаёт результаты пользователю.

Здесь используется упрощённый подход: PromoterStats не ищет модуль целиком, а выполняет поиск отдельных факторов, которые статистически встречаются чаще (или реже) на заданной пользователем выборке промоторов (по сравнению со всем набором промоторов, использованных в данном эксперименте). Для статистической оценки используется z-тест [8]. Оценивается среднее количество сайтов связывания каждого фактора на промоторе, а также количество промоторов, где присутствует хотя бы один сайт связывания. Задача решается за линейное время от числа промоторов и числа факторов, результат получается сразу же после отправки формы с исходными данными.

3. COMPOSITE MODULE ANALYST

Composite Module Analyst (CMA), разрабатываемый нашей группой, написан на C++ и на данный момент представляет приложение с интерфейсом командной строки, компилируемое под Win32 и Unix. Планируется также разработать веб-интерфейс и Win32 GUI. Он включает в себя два этапа: поиск сайтов связывания и поиск модуля, причём этапы могут быть выполнены по отдельности (с сохранением промежуточных результатов) или вместе.

Здесь на вход подаётся либо два набора промоторов, либо набор промоторов и соответствующие им численные значения экспрессии. В первом случае один из наборов включает промоторы с повышенной экспрессией, а второй — остальные промоторы, использованные в эксперименте, и выполняется поиск такого модуля, который даёт большой вес на первом наборе и малый на втором. Во втором случае учитывается также величина экспрессии и ищется такой модуль, веса которого для каждого промотора лучше всего соответствуют величинам экспрессии. Строится зависимость величины веса промотора от его экспрессии, аппроксимируется прямой и вычисляется квадрат смешанной корреляции для аппроксимации, который и максимизируется.

Поиск лучшего модуля осуществляется генетическим алгоритмом, его реализация похожа на реализацию в TOUCAN. Кроме поиска обычных модулей, состоящих из заданного числа факторов, реализован также поиск так называемого булева модуля. Булев модуль представляет собой булеву формулу следующего вида:

$$\overline{\left(\bigcup_i p_{0i}\right)} \cap \left(\left(\bigcap_j \bigcup_i p_{ij}\right)\right).$$

Здесь p_{ij} — логическая величина, определяющая, есть ли на данном промоторе сайты связывания определённого фактора. Формула представляет собой конъюнкцию дизъюнкций, причём первый конъюнкт с отрицанием. С точки зрения биологии это означает, что факторы, вошедшие в первый конъюнкт, препятствуют повышению экспрессии, а факторы, оказавшиеся внутри одного конъюнкта, взаимозаменяемы. Число конъюнктов, дизъюнкций внутри конъюнкта и общее число матриц могут варьироваться в заданных пределах. Такие модули также могут мутировать и скрещиваться, как и обычные.

Помимо нахождения лучшего модуля, CMA позволяет решать некоторые сопутствующие задачи: кластеризацию входного набора промоторов (разделение набора на подмножества, к которым найденный модуль подходит лучше всего), прогон алгоритма несколько раз и сравнение результатов для тестирования устойчивости, подсчёт веса конкретного модуля, заданного пользователем, и другие.

4. СРАВНЕНИЕ ПРОГРАММНЫХ ПАКЕТОВ

Три рассмотренных разработки обладают своими достоинствами и недостатками. Система TELiS больше отличается от остальных: в ней не используется модульный подход, а рассматривается каждый фактор по отдельности. Основной недостаток такого подхода в том, что не учитывается расстояние между сайтами связывания на промоторе. Отчасти эта проблема решена: можно выполнить расчёт либо для 300 первых нуклеотидов промотора, либо для 600 или 1200. Однако этого параметра недостаточно, сайты связывания конкретного комплекса могут располагаться, скажем, с 200-го по 400-й нуклеотид. С другой стороны, такое упрощение сводит задачу к линейной сложности и позволяет быстро найти факторы, которые потенциально могут присутствовать в оптимальном модуле. Кроме того, TELiS не ограничивает число найденных факторов, тогда как TOUCAN и CMA ищут

модули с конкретным числом факторов, хотя это число может быть заранее неизвестно, и придётся прогонять алгоритм, меняя эту величину.

Система TOUCAN реализует алгоритм, который находит заведомо лучший модуль. Несмотря на то, что он выполняется долго и требует много памяти, иногда может быть полезно получить гарантированно оптимальный результат. Хотя, как показывает практика, генетический алгоритм, реализованный и в TOUCAN, и в CMA, также даёт хорошие результаты за гораздо меньшее время.

Система TOUCAN имеет некоторые дополнительные возможности, отсутствующие в CMA: возможность искать модули с повторяющимися факторами или запрет поиска модулей, где сайты факторов перекрываются. Кроме того, в TOUCAN для пар факторов введена степень сходства, и можно запретить наличие в модуле факторов со степенью сходства выше некоторой заданной величины.

TELiS и TOUCAN реализованы на Java, что упрощает переносимость этих систем, но значительно снижает быстродействие. В частности, это стало причиной того, что поиск сайтов связывания разработчики TELiS выполнили заранее для всевозможных генов и матриц, упомянув, что эти вычисления могут занять несколько дней. Та же процедура в CMA для аналогичных объёмов входных данных занимает около часа. Реально же в каждом эксперименте не требуется информация про абсолютно все гены, поэтому поиск сайтов для конкретного эксперимента может занимать меньше минуты. Следует также заметить, что полученная база данных TELiS не содержит информации о положении сайтов связывания, а содержит лишь количество сайтов определённого фактора для промотора каждого гена. Это делает невозможным использование её в других разработках, где необходимо учитывать расстояние между сайтами.

Трудно сравнить быстродействие TOUCAN и CMA, так как реальные вычисления, выполняемые TOUCAN, производятся на кластере разработчиков. Тем не менее, CMA демонстрирует вполне удовлетворительное быстродействие на одном компьютере класса Pentium III, необходимости переносить вычисления на кластер нет. В настоящий момент программа CMA используется в рамках работы по гранту INTAS «Построение модели регуляторной сети в нормальном и патологическом состоянии для предсказания потенциальных противораковых фармакологических агентов для ключевых молекул» для выявления комплексов факторов, регулирующих работу генов клеточного цикла в зависимости от его стадии.

ЗАКЛЮЧЕНИЕ

В статье проведён обзор различных систем, предназначенных для поиска наборов транскрипционных факторов. Каждая из рассмотренных систем обладает своими достоинствами и недостатками и может быть полезна в определённых ситуациях. Система TELiS полезна, если необходимо узнать, какие факторы повышают экспрессию в данном эксперименте, и не так важно, рядом расположены их сайты или нет. Пакет TOUCAN позволяет найти оптимальный модуль для небольших наборов данных, а также выполнить генетический алгоритм. Систему CMA можно использовать для достаточно больших наборов входных данных, выполнять поиск булева модуля и решать смежные задачи типа кластеризации набора промоторов.

Дальнейшее исследование этой области неизбежно. Планируется более точное математическое моделирование биологических процессов, происходящих при регуляции генов, и программная реализация этих моделей. Также будет исследована возможность усовершенствования генетического алгоритма с целью ускорения поиска модулей с большим числом факторов, когда количество возможных модулей резко возрастает. Кроме того, планируется ввести новые статистические тесты для оценки надёжности полученного результата.

СПИСОК ЛИТЕРАТУРЫ

1. **Velculescu V. E., Zhang L., et al.** Serial analysis of gene expression // Science. — 1995. — N 270 (5235). — P. 484–487.
2. **Aerts S., Thijs G., Coessens B., et al.** TOUCAN: Deciphering the Cis-Regulatory Logic of Coregulated Genes // Nuclear Acids Research. — 2003. — Vol. 31, N 6 — P. 1753–1764.
3. **Cole S., Yan W., Galic Z., et al.** Expression-based monitoring of transcription factor activity: The TELiS database // Bioinformatics. — 2005. — N 21 (6). — P. 803–810.
4. **Konovalova T., Cheremushkin E., Beschastnov E., Kel. A.** Applicating of the metropolis algorithm to reveal composite modules in promoters of eukaryotic genes // Proc. European Conf. on Computational Biology, Paris, France, September 2003. — Paris, 2003. — P. 447–448.
5. **Aerts S., Van Loo P., Thijs G., et al.** Computational detection of cis-regulatory modules // Bioinformatics. — 2003. — Vol. 19, Suppl. 2. — P. ii5–ii14.
6. **Land A.H., Doig A.G.** An automatic method of solving discrete programming problems // Econometrica. — 1960. — Vol. 28 — P. 497–520.

7. **Aerts S, Van Loo P, Moreau Y, De Moor B.** A genetic algorithm for the detection of new cis-regulatory modules in sets of coregulated genes // Bioinformatics. — 2004. — Vol. 20, N 12. — P. 1974–1976.
8. **Kanji G. K.** 100 Statistical Tests. — London, Sage, 1999. — 224 p.