Исправление ошибок в чтениях, полученных с помощью технологии IonTorrent

Ершов В. А., студент кафедры статистического моделирования СПБГУ, vasilij.ershov@gmail.com

Коробейников А. И., к.ф.-м.н., доцент кафедры статистического моделирования СПБГУ, anton@korobeynikov.info

Аннотация

Для того, чтобы изучать ДНК организмов требуется каким-либо образом преобразовать макромолекулу в удобный для анализа формат. Для широкого спектра задач этим форматом является строчка из алфавита $\{A, C, G, T\}$. Современные технологии секвенирования позволяют преобразовать в такой формат лишь небольшие участки ДНК. При этом, получающиеся строчки содержат ошибки, которые требуется исправлять. В данной работе мы разрабатываем алгоритм исправления ошибок, возникающих при использовании технологии IonTorrent.

Введение

В биологии и медицине изучение ДНК организмов возникает в огромном количестве задач. Но для того, чтобы биологи смогли изучать ДНК организма, его требуется секвенировать — преобразовать из макромолекулы в строчку из алфавита $\{A, C, G, T\}$. Современные технологии позволяют секвенировать только большое количество небольших участков ДНК. Прочитанный участок ДНК мы будем называть чтением. К сожалению, в процессе секвенирования возникают ошибки. Для большинства приложений эти ошибки требуется так или иначе исправлять. В частности, исправление ошибок позволяет улучшить качество сборки генома — получения полной цепочки ДНК на основе тех участков, которые удалось прочитать.

Существует несколько технологий секвенирования. Одними из самых распространенных являются Illumina и IonTorrent. Для технологии Illumina в геномном ассемблере SPAdes[2] реализован алгоритм BayesHammer[1]. Для технологии IonTorrent реализован аналог данного алгоритма — IonHammer[4]. Текущая реализация не полностью использует особенности технологии IonTorrent: в некоторых частях алгоритма потенциально возможны существенные улучшения, разработкой которых мы и занимаемся.

Основная идея алгоритма исправления ошибок

Пусть $\mathbb{A} = \{A, C, G, T\}$ — множество возможных нуклеотидов. В результате работы секвенсора IonTorrent получается множество чтений $\mathbb{S}$, где $s \in \mathbb{S}$ — некоторая строчка из элементов $\mathbb{A}$. Особенность технологии IonTorrent состоит в том, что за раз читается не один нуклеотид, а один гомополимер, т.е. несколько идущих подряд нуклеотидов с одинаковым основанием. Разумно ожидать, что большая часть ошибок — неверная оценка длины гомополимера. Гомополимеры будем обозначать как пару (n, l), где $n \in \mathbb{A}$ — нуклеотид, а $l \in \mathbb{N}$ — его длина. Множесто всех возможных гомополимеров будем обозначать за $\mathbb{H}$. hk-мером будем называть элемент $\mathbb{H}^k$.

Ясно, что множество чтений $\mathbb{S}$ для любого k порождает множество встретившихся в чтениях hk-меров. Одно из предположений, на котором основан алгоритм IonHammer заключается в том, что для исправления ошибок в чтениях достаточно для некоторого k научиться оценивать множество геномных hk-меров (т.е. hk-меров, содержащихся в ДНК). Для дальнейшего описания алгоритма нам потребуется ввести следующее определение:

Определение. Пусть $d(x, y)$ — некоторая «мера различия» между двумя hk-мерами (отображение из $\mathbb{H}^2$ в $\mathbb{R}_{\geq 0}$). Тогда ED_l-графом (англ. Edit distance graph) называется граф, вершинами которого являются hk-меры и между двумя вершинами x и y есть ребро тогда и только тогда, когда $d(x, y) \leq l$.

Метод оценивания множества геномных hk-меров в IonHammer основан на кластеризации вершин ED_l-графа. Предположим, пока, что у нас зафиксирована «мера различия» d и выбрано $k \in \mathbb{N}$. Тогда в первом приближении алгоритм оценки геномных центров в IonHammer можно представить следующим образом:

1. По всем чтениям считаем аддитивные статистики для всех встречающихся hk-меров и получаем множество $\{s_x\}$, где x — hk-мер, а s — некоторая статистика (например, количество раз, которое встретился данный hk-мер).

2. Строим ED_l-граф.

3. Разбиваем ED_l-граф на компоненты связности. Не умаляя общности будем считать, что весь граф является одной компонентой связности.

4. На основе посчитанных статистик с помощью некоторого алгоритма кластеризации разбиваем вершины графа на кластера.

5. Фиксируем центры получившихся кластеров.

6. С помощью некоторой «метрики качества» выбираем те «центры», которые мы «готовы считать геномными» (т.е. не содержащими ошибок).

Выделим важные предположения, без которых данный алгоритм использовать не получится:

- Ошибочные hk-меры не содержатся в множестве геномных hk-меров (которое мы не знаем и будем пытаться оценить).
- Ошибочные hk-меры будут «близки» к истинным в ED_l графе.

В итоге, для успешного исправления ошибок требуется правильно задать «параметры» алгоритма — расстояние d, длину hk-меров k, параметр ED_l графа l, «метрику качества» hk-меров и проверить, что обозначенные выше предположения выполнены. Для оптимального исправления ошибок параметры требуется выбирать, учитывая специфику работы IonTorrent.

Статистический профиль ошибок

Таким образом, для того, чтобы понять, как правильно исправлять ошибки, требуется узнать, какие ошибки, как, когда и насколько часто совершаются. Отметим также, что данная информация также важна и для следующего шага алгоритма — исправления ошибок в чтениях на основе геномных hk-меров.

Для ответа на возникшие вопросы мы выбрали несколько наборов чтений для бактерий *E. coli str. K12* и *E. coli str. DH10B*, доступных с сайта IonCommunity [3]. Основной нашей задачей стояло проверить то, насколько разумно выбраны параметры в текущей реализации IonHammer. В результате мы выяснили, что:

- Расстояние Левенштейна и расстояние Хэмминга на hk-мерах удовлетворяют необходимым предположениям.
- Достаточно рассматривать hk-меры с $k = 16$. При таком выборе k ошибочные hk-меры в основном не будут принадлежать множеству геномных, при этом они будут «близки» к геномному hk-меру, не содержащему допущенных ошибок.
- Большая часть ошибочных hk-меров находится на расстояние 1 от «истинного» hk-мера. Таким образом, достаточно рассматривать ED_1-граф для кластеризации.

- На основе дополнительной статистики о качестве прочитанных нуклеотидов, доступной в результате работы IonTorrent, можно достаточно точно определять геномные hk-меры — можно задать такую «метрику качества», что вероятность отнести негеномный hk-мер к геномному будет менее 0.001, при этом при этом среди отнесенных к плохим центрам будет менее 0.001% геномных.

Таким образом, мы поняли, что предположения, на которых основан алгоритм оценивания геномных центров в IonHammer можно считать выполненными. Как-либо менять текущие параметры не требуется. Также мы поняли, какие центры, полученные в результате кластеризации, можно считать геномными. Кроме того, информация о том, какие ошибки совершает IonTorrent, активно используется в реализованном нами алгоритме исправления ошибок на основе оцененного множества геномных hk-меров.

Исправление ошибок на основе геномых hk-меров

После того, как оценено множество геномных hk-меров, остается с его помощью исправить ошибки в чтениях. Для решения данной задачи мы реализовали новый алгоритм, являющийся адаптацией алгоритма коррекции из BayesHammer и активно использующий особенности секвенирования с помощью технологии IonTorrent. В отличии от текущего алгоритма коррекции, новый алгоритм более, чем в 5 раз быстрее и лучше исправляет ошибки. В частности, новый алгоритм практически не ухудшает чтения, в отличии от предыдущего.

Пусть требуется исправить чтение s. Будем считать, что s является последовательностью гомополимеров, т.е. $s[i] \in \mathbb{H}$, где под $s[i]$ мы обозначем i-ый «символ» строки. Под $s[i{:}j]$ будем понимать подстроку из гомополимеров с позиции i до позиции j.

Предположим, что $s[1{:}k]$ является геномным hk-мером. Рассмотрим оптимальный алгоритм коррекции чтения s:

- Пусть $\tilde{s} = s[1{:}k]$.
- Будем добавлять в $\tilde{s}$ гомополимеры из s. Пусть i — позиция первого гомополимера из s, который еще не добавлен в $\tilde{s}$.
- Операцию объединения двух строчек будем обозначать точкой. Тогда до тех пор, пока $(\tilde{s}.s[i])[(i-k){:}i]$ является геномным hk-мером, обновляем $\tilde{s} = \tilde{s}.s[i]$ и увеличиваем i

- Пусть теперь добавление $s[i]$ приводит к негеномному hk-меру. Тогда наши исследования статистического профиля показывает, что естественно предположить, что при чтении $s[i]$ произошла ошибка. Для ее исправления переберем все возможные ошибки, которые мог допустить IonTorrent в данном гомополимере. Для всех возможных коррекций введем некоторый штраф. Рекурсивно запустим этот же алгоритм: для каждой коррекции, которая приводит к геномному hk-меру, добавим в $\tilde{s}$ исправленный гомополимер и увеличим i на один.

- В результате получим множество различных коррекций s. Выберем ту из них, которая имеет наименьший штраф.

Данный алгоритм действительно является оптимальным (при предположении, что у нас есть доступ к истинному множеству геномных hk-меров) — без дополнительных ограничений он переберет все возможные коррекции чтения s. В то же время, алгоритм очень трудоемкий — количество возможных коррекций достаточно часто будет расти экспоненциально. Используемый нами алгоритм является аппроксимацией данного оптимального алгоритма. Во-первых, за счет изучение статистического профиля ошибок мы знаем, что все возможные ошибки перебирать не требуется — основные ошибки, которые допускает IonTorrent — вставки и удаления одного нуклеотида. Кроме того, большая часть ошибочных hk-меров близка к геномным. На основе этих наблюдений мы добавили набор эвристик, позволяющие отсекать маловероятные коррекции.

Для использования данного алгоритма также требуется правильно выбрать функцию штрафа. Данный выбор может существенно влиять на качество исправления ошибок. В текущей реализации для штрафов используется очень грубые оценки логарифма вероятности того, что IonTorrent допустил ту или иную ошибку. При этом мы ни как не используем информацию о том, на какой позиции мы делаем исправление, хотя количество ошибок в конце чтений существенно больше, чем в начале, а также некоторые ошибки находятся «близко» друг от друга (основным примером такой ошибки является пропуск нуклеотида, а затем вставка его же в следующий гомополимер с соответствующим основанием). Поэтому следующим шагом мы планируем реализовать автоматическое оценивание функции штрафа на основе ED_l-графа.

Заключение

В данной работе рассмотрен алгоритм IonHammer, предназначенный для коррекции ошибок в чтениях, полученных с помощью секвенсора IonTorrent.

Мы проанализировали, какие ошибки допускает IonTorrent на реальных данных и проверили корректность параметров, влияющих на качество работы IonHammer. Также мы предложили новый алгоритм исправления ошибок, превосходящий по скорости и качеству использующийся сейчас алгоритм коррекции.

Литература

[1] Sergey I. Nikolenko, Anton I. Korobeynikov and Max. A. Alekseyev. BayesHammer: Bayesian clustering for error correction in single-cell sequencing. BMC Genomics (2013) 14(S1):S7. doi:10.1186/1471-2164-14-S1-S7

[2] Sergey Nurk, Anton Bankevich, Dmitry Antipov, Alexey A. Gurevich, Anton Korobeynikov, Alla Lapidus, Andrey D. Prjibelski, Alexey Pyshkin, Alexander Sirotkin, Yakov Sirotkin, Ramunas Stepanauskas, Scott R. Clingenpeel, Tanja Woyke, Jeffrey S. Mclean, Roger Lasken, Glenn Tesler, Max A. Alekseyev, and Pavel A. Pevzner. Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. Journal of Computational Biology. October 2013, 20(10): 714-737. doi:10.1089/cmb.2013.0084.

[3] IonCommunity — URL: https://ioncommunity.thermofisher.com (online; accessed 5.04.2016)

[4] Anton Korobeynikov, Artem Tarasov, Alla Lapidus, Pavel A. Pevzner — IonHammer: homopolymer-space Hamming clustering for IonTorrent read error correction (unpublished).