Зачем биологам синтаксический анализ

Автор: Артём Горохов

Санкт-Петербургский государственный университет
Лаборатория языковых инструментов JetBrains

15 октября 2016г.

Биоинформатика

- Множество задач, связанных с обработкой и пониманием биологических данных
- Одна из задач - поиск организмов в метагеномных сборках

Геном

- Геном (последовательность ДНК) - длинная последовательность нуклеотидов
- На деле строка над алфавитом {A, C, G, T}

Секвенирование

- Из биологического материала получаются риды - последовательности строчек длиной около 100 символов
- Риды склеиваются в более длинные строки
- Множество полученных строк - сборка

Сборки

В зависимости от биологического материала, из которого получаются данные, сборки бывают:

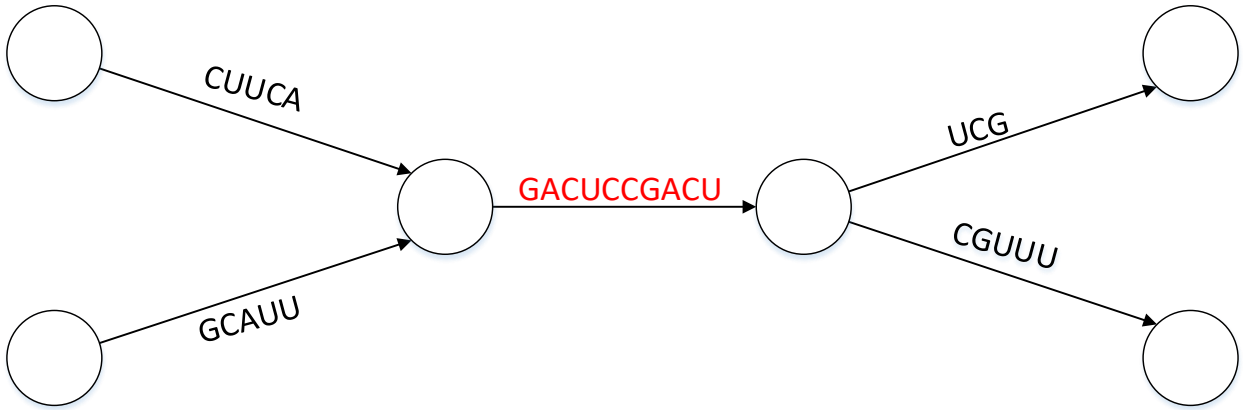
- single-cell. Сборки, для получения которой была взята одна или несколько (до четырёх) клеток колонии
- multi-cell. В качестве биологического материала были взяты тысячи или даже десятки тысяч клеток одного штамма
- метагеномные. Данные взяты из среды обитания целевой колонии, в которой были как её представители, так и соседствующих

Метагеномная сборка

CUUCAGACUCCGACUUCG

UCCGACUCGUUU

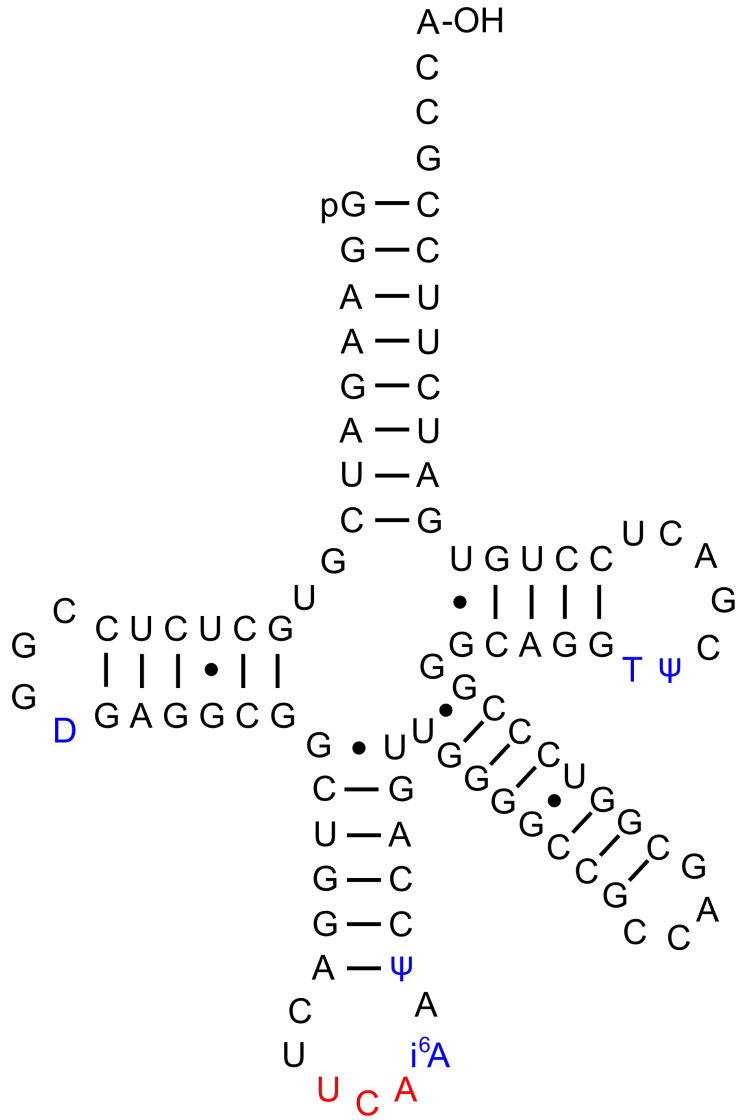
GCAUUGACUC



Что ищем

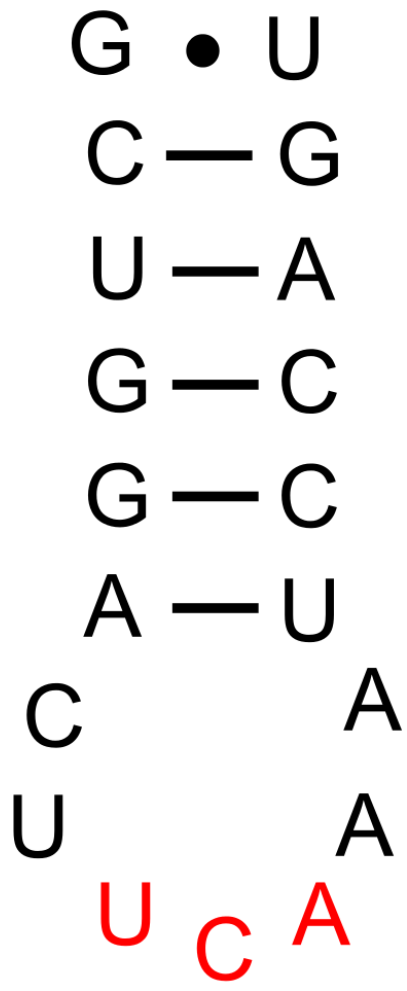
- Такие последовательности как тРНК, рРНК могут определить организм которому они принадлежат
- Есть у этих последовательностей есть вторичная структура, которая может быть описана КС-грамматикой

$GGAAGAUCG...GCA... =>$

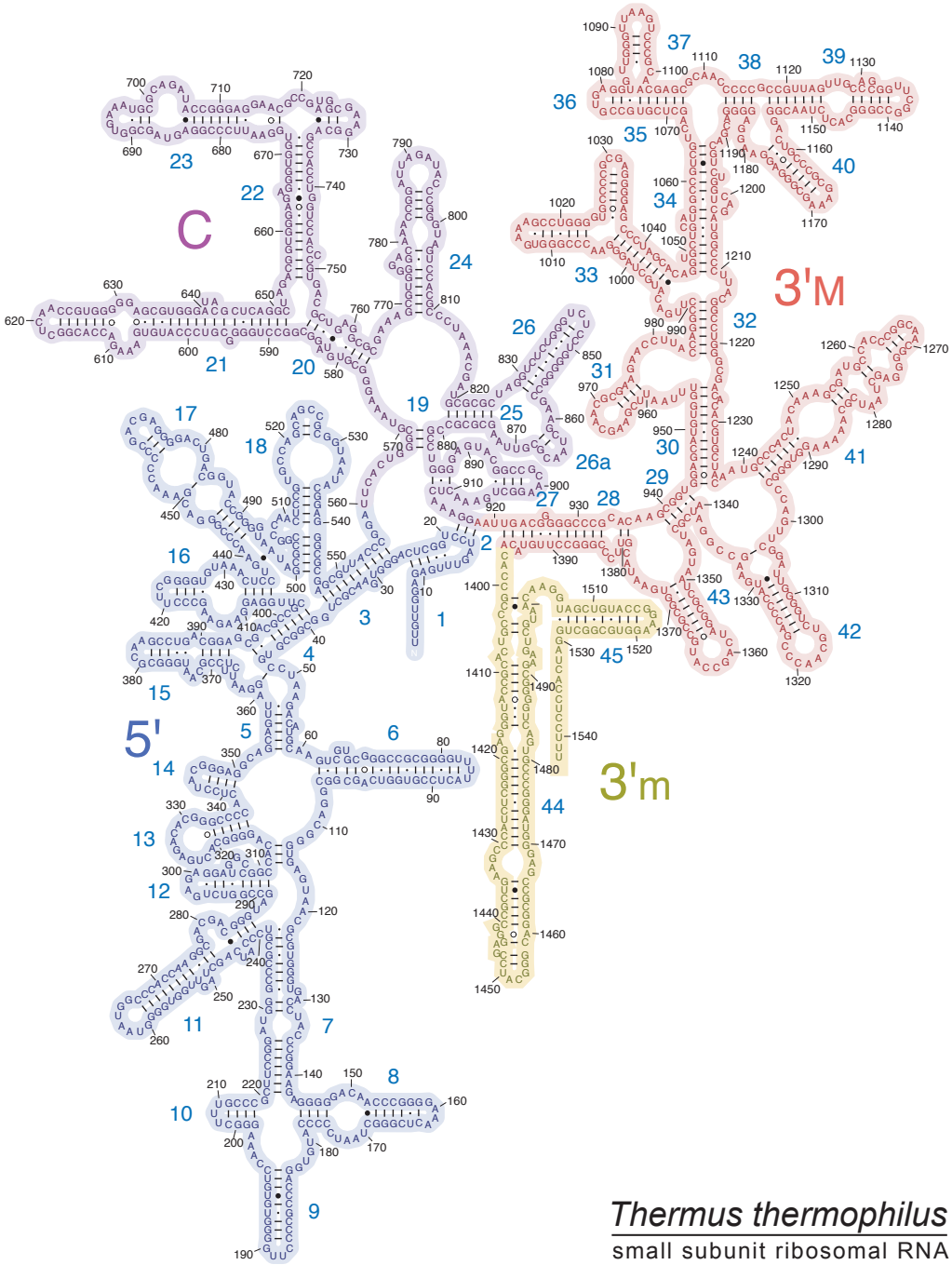


Грамматика для кусочка тРНК

$$
\begin{aligned}
START &= STEM \\
STEM &= a\ STEM\ u \\
&\mid u\ STEM\ a \\
&\mid c\ STEM\ g \\
&\mid g\ STEM\ c \\
&\mid g\ STEM\ u \\
&\mid u\ STEM\ g \\
&\mid ANY^*[3..6] \\
ANY &= a \mid u \mid g \mid c
\end{aligned}
$$



Вторичная структура
16s рРНК



GLL

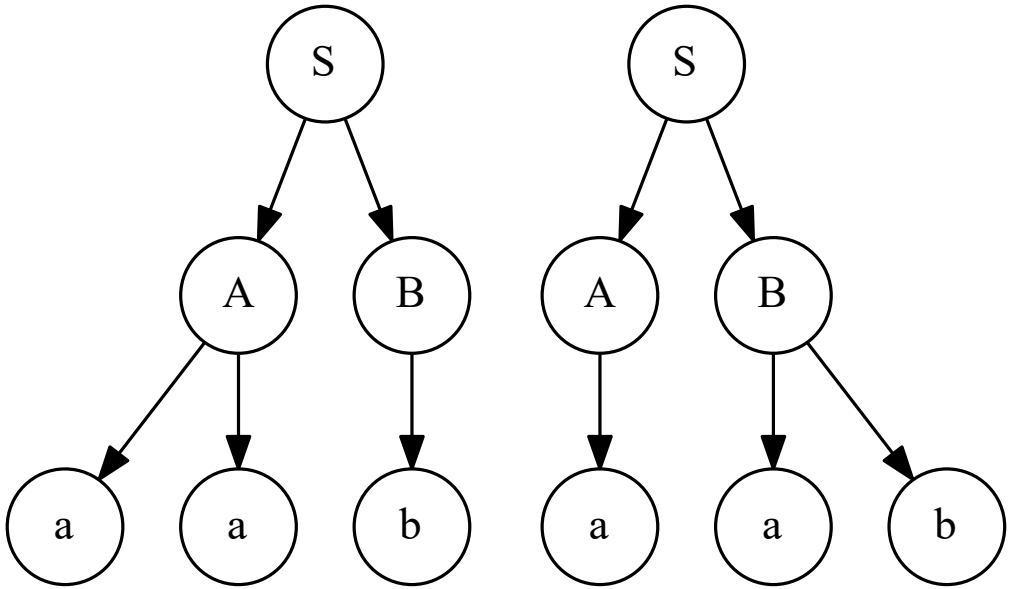
- Generalized LL
- Нисходящий синтаксический анализатор
- В лучшем случае работает за линейное время, в худшем - за $O(n^3)$
- Строит все возможные выводы цепочки

GLL

Вход: a a b

Грамматика:

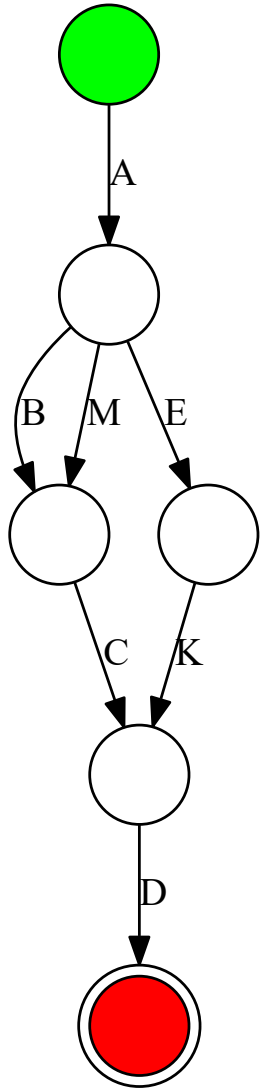
$$
\begin{aligned}
S &= A\ B \\
A &= a\ a \\
&\mid a \\
B &= b \\
&\mid a\ b
\end{aligned}
$$



GLL для графов

- На вход поступает граф, задающий все входные цепочки
- На рёбрах терминалы

$$\{ABCD; AMCD; AEKD\} =>$$



Начальные результаты

Полученные метагеномные сборки не поддаются анализу без предварительных преобразований

Увеличение производительности

- Необходимо подготавливать сборки к синтаксическому анализу
- Сам алгоритм нуждается в модернизации

Фильтрация рёбер

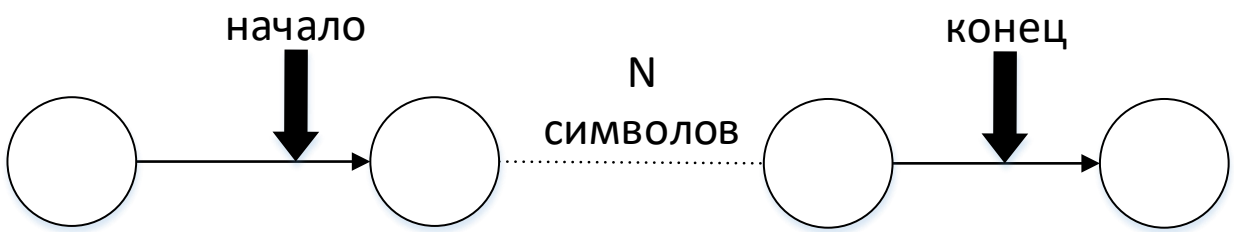
- Infernal позволяет распознавать структуры в линейном входе
- Рёбра, длиннее искомых структур можно делить на части и проверять infernal'ом

Разбиение на компоненты

- После фильтрации рёбер граф распадается на компоненты связности
- Можно запускать анализатор независимо на разных компонентах

Отказ от построения дерева

- Парсер возвращает лишь границы и длину найденной цепочки
- Восстановление цепочки идёт путём извлечения подграфа
- Ложные фильтруются infernal



Преобразование грамматики

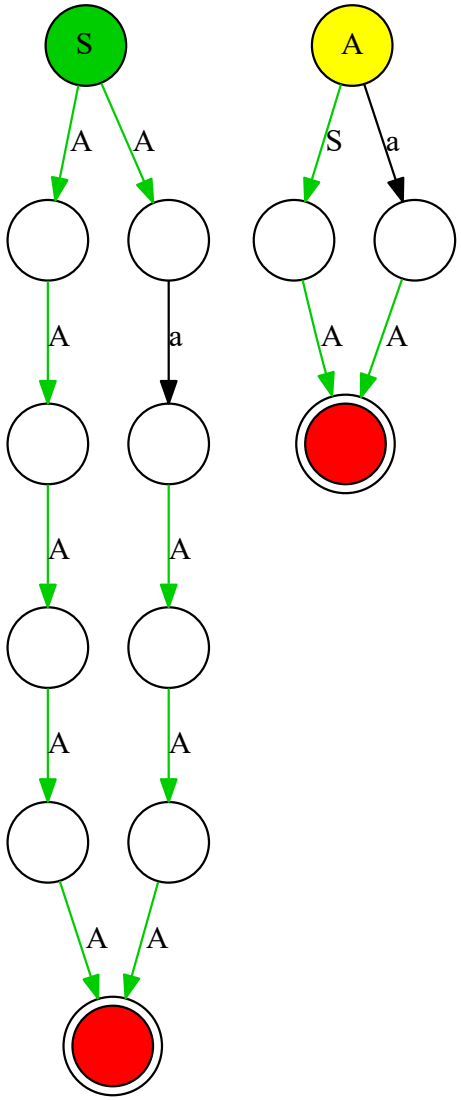
- Грамматика для 16s сильно неоднозначная и довольно большая
- Можно минимизировать её

Преобразование грамматики к автомату

Грамматика

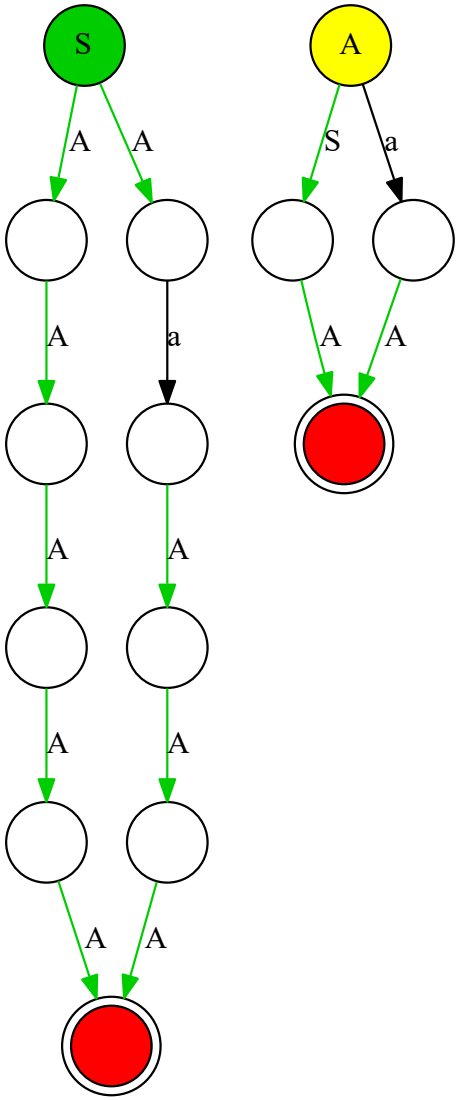
$$
\begin{aligned}
S &= A\ A\ A\ A\ A \\
&\mid A\ a\ A\ A\ A \\
A &= S\ A \\
&\mid a\ A \\
&\mid a
\end{aligned}
$$

Автомат

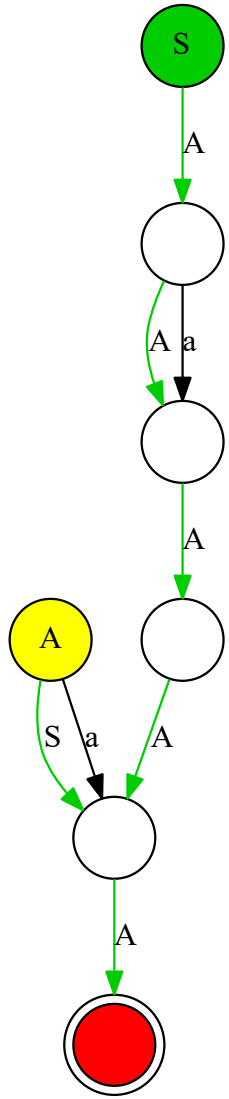


Минимизация автомата

Изначальный автомат



Минимизированый автомат



Эксперименты

	начальная грамматика	мин. автомат
Время работы	10 часов	3ч. 40 мин.

Направление работ

- Детальный анализ качества результата
- Возможно, можно сильнее фильтровать граф, применяя infernal
- Поиск полноразмерных 16s
- Поиск других структур