Разработка алгоритмов обработки графа де Брюина в задаче геномного ассемблирования

Нурк Сергей Юрьевич, 545 гр.

Кафедра системного программирования

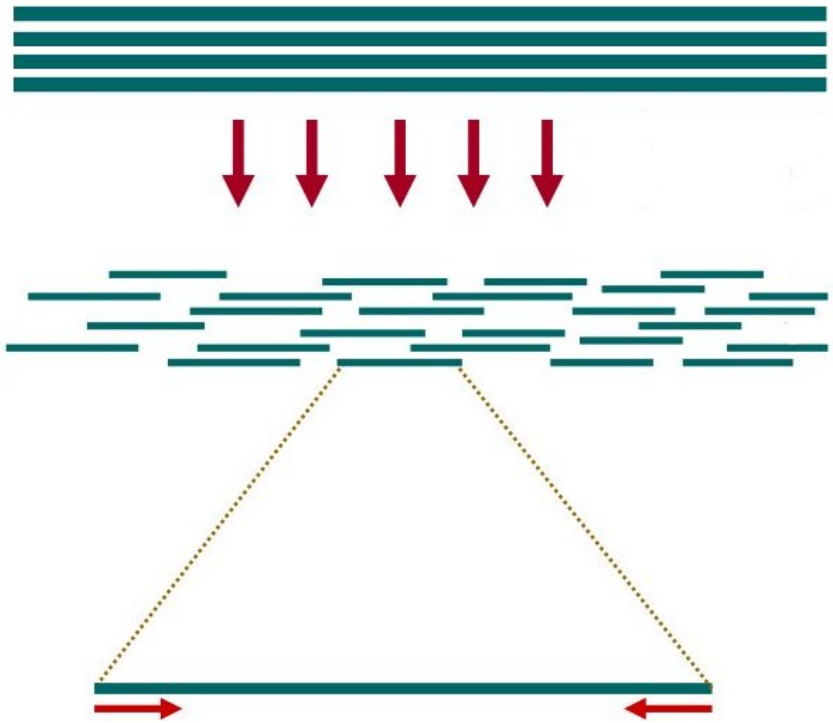
Научный руководитель: Н.И. Вяххи

Рецензент: М.А. Алексеев

Мат-Мех СПбГУ
1 июня 2011г.

Геном и секвенирование

- Геном закодирован в ДНК. Алфавит из 4 букв (A,T,G,C).
- *E.coli* — 4.5 млн. букв, человек — 3.2 млрд. букв.
- Секвенирование — процесс определения генома.
- Результат — короткие парные фрагменты $(100 - 200$ нуклеотидов$)$.
- Ассемблинг — восстановление исходной последовательности



Граф де Брюина

Определение

Для фиксированного k, алфавита M и множества строк $S \subset M^k$ $G_k(S) =< V, E >$, где

- $V = \{v \in M^{k-1} | \exists s \in S : v = start(s) \vee v = end(s)\}$
- $E = \{(v_1 \to v_2) | \exists s \in S : v_1 = start(s) \wedge v_2 = end(s)\}$

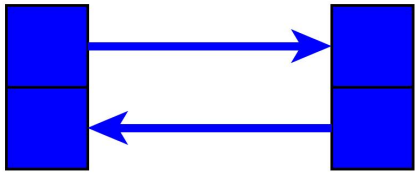
Сжатый граф де Брюина

Получается заменой каждого неразветвленного участка одним ребром. На ребрах оказываются последовательности произвольной длины.

Геном соответствует некоторому пути в графе.

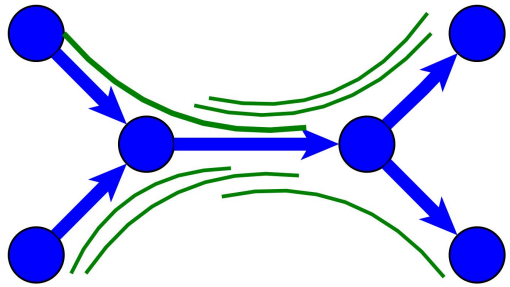
Дополнительные свойства графа

- Сопряженная структура
- Покрытие ребер
- Межреберные расстояния



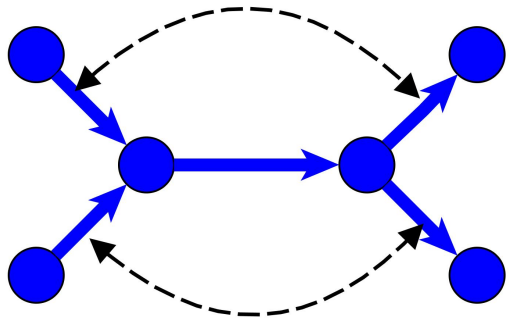
Дополнительные свойства графа

- Сопряженная структура
- Покрытие ребер
- Межреберные расстояния



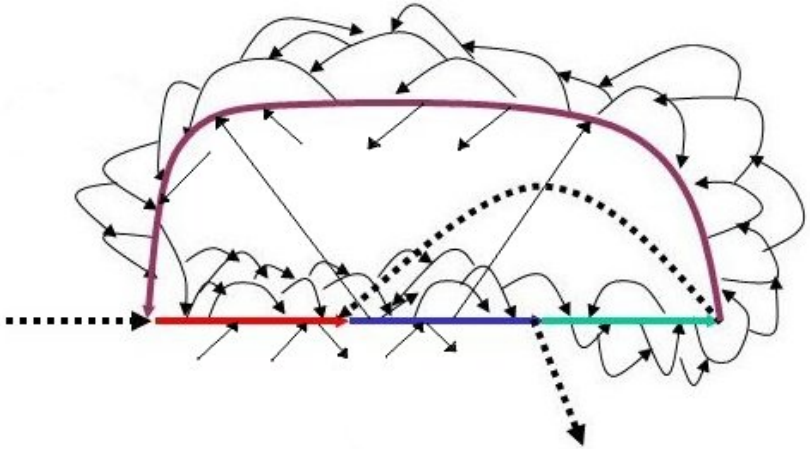
Дополнительные свойства графа

- Сопряженная структура
- Покрытие ребер
- Межреберные расстояния



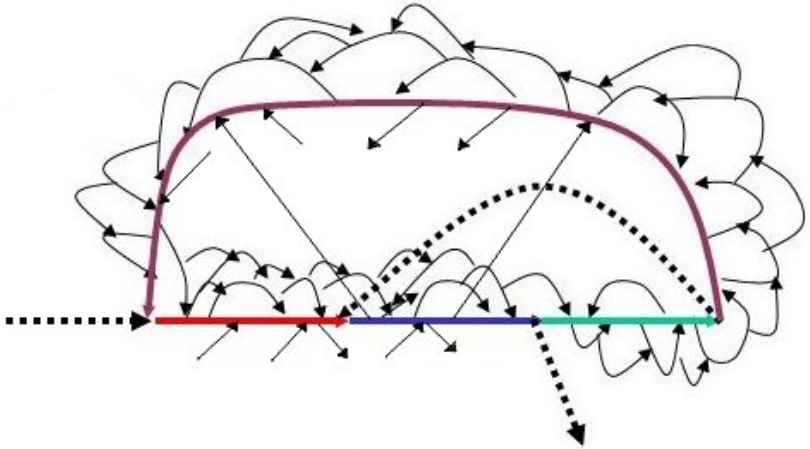
Упрощение графа

Граф, по фрагментам

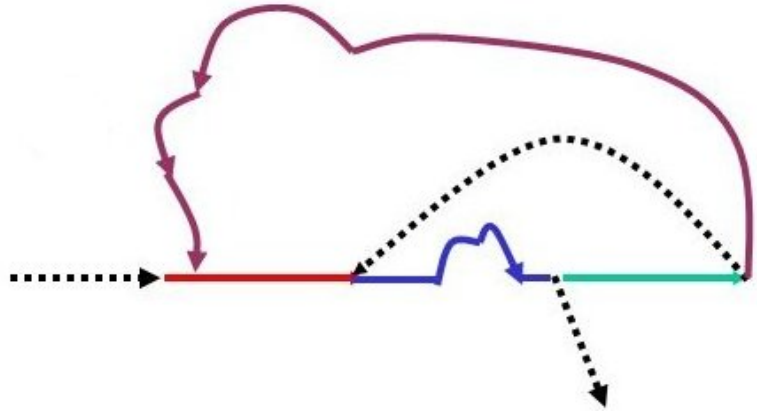


Упрощение графа

Граф, по фрагментам



Упрощенный граф



Основные цели работы

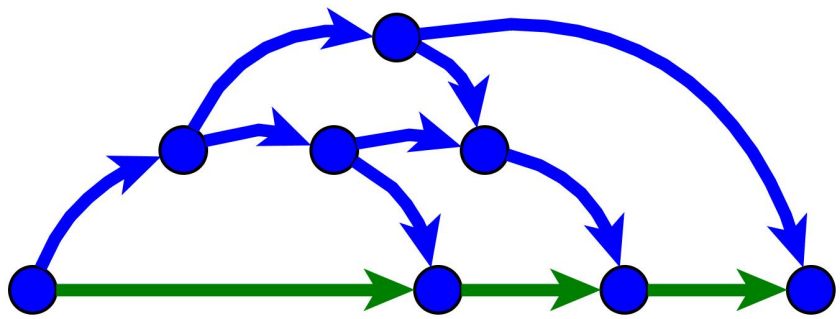
- Спроектировать архитектуру для удобной работы с графом.
- Разработать алгоритмы устранения ошибочных участков графа.

Особенности реализованной архитектуры

- Введены операции преобразования графа, учитывающие сопряженную структуру.
- Разработаны методы пересчета покрытия и межреберных расстояний в соответствии с семантикой операций преобразования.
- Постоянно поддерживается корректность дополнительной информации.
- Поддержание корректности “прозрачно” для алгоритмов обработки.
- **Итог:** просто добавлять новые типы информации, просто реализовывать алгоритмы обработки.

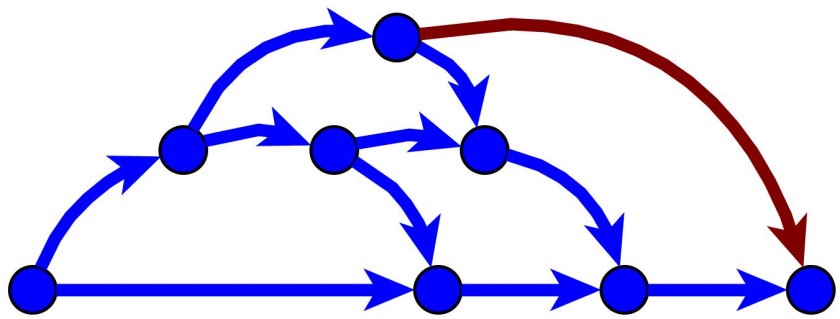
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



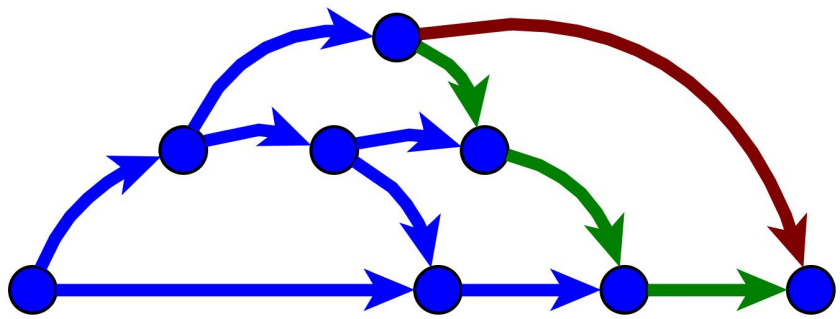
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



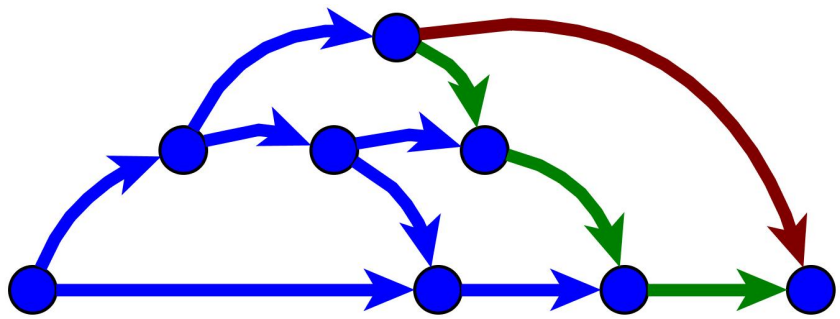
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



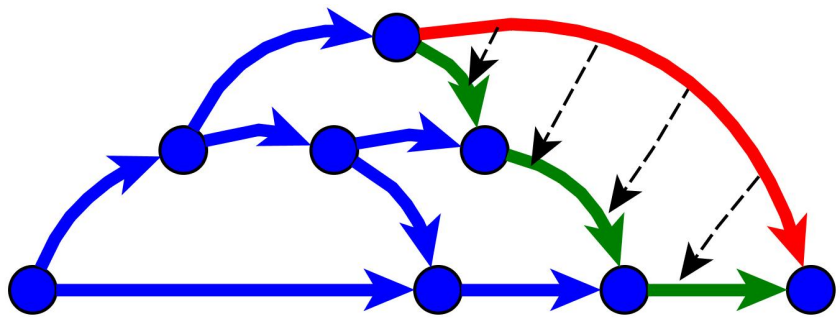
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



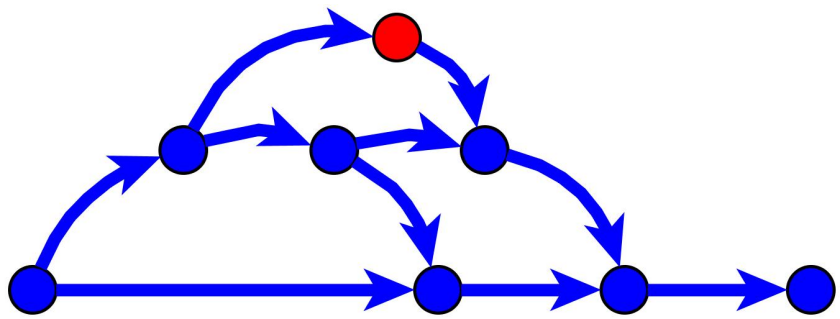
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



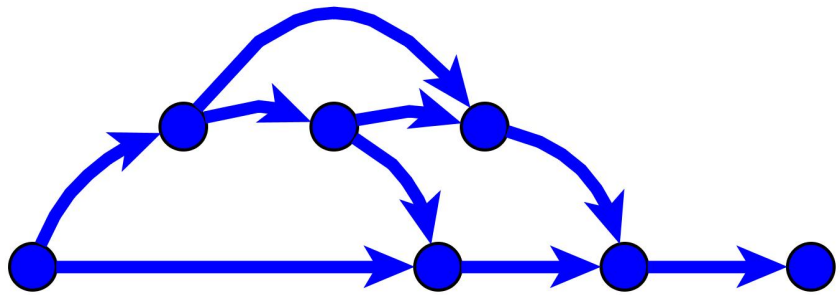
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



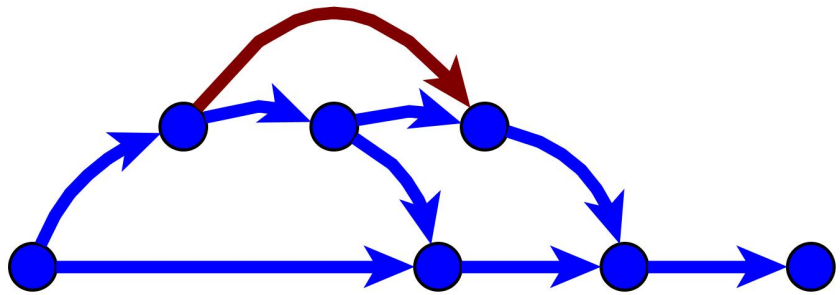
Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



Алгоритм устранения “пузырей”

- Для каждого достаточно короткого ребра e
- Если есть альтернативный путь, от начала к концу, то:
- Если для него выполняются некоторые условия, то:
- Удалим e, “спроектировав” на альтернативный путь.
- Сожмем окрестности концов и включим новые ребра в обход.



Особенности алгоритмов

- Возможность тонкой настройки.
- Возможность не использовать последовательности нуклеотидов за счет использования покрытия и межреберных расстояний.
- На реальных данных по *E.coli* удалено 98% ошибочных ребер.
- Сложность в текущей реализации $O(|E|log|E|)$.

Итоги

- Разработаны алгоритмы построения и сжатия графа де Брюина.
- Разработаны алгоритмы устранения основных типов ошибочных участков графа.
- Разработаны новые методы работы с покрытием и межреберными расстояниями.
- Архитектура позволяет легко расширять функциональность и разрабатывать новые алгоритмы.
- Библиотека реализована на C++ в рамках проекта лаборатории алгоритмической биологии СПбАУ РАН.