Кластеризация Хэмминг графа, построенного по данным секвенирования антител

Руководитель: Яна Сафонова

Дешевая рабочая сила: Миша Никонов

Задача

Восстановить последовательности антител и их кратности из репертуара. Проблема в том, что секвенирование добавляет ошибки, которые нужно исправить, не испортив естественные вариации. Для этого мы строим Хэмминг граф на всех последовательностях репертуара и пытаемся его кластеризовать так, чтобы каждый кластер соответствовал правильному антителу.

Алгоритмическая постановка

Нужно кластеризовать граф, который обладает следующими свойствами:

- Кластеры - это почти клики.
- Между кластерами много ребер всевозможных весов.
- Максимальный вес ребра равен 3-м.
- С вероятностью где-то 10% расстояние между двумя вершинами, которые относятся к одной компаненте, больше нуля.
- Граф необязательно связан.
- Количество вершин не более 10^6, но пока ограничемся диапозоном от 100 до 35000(5Gb).

Существующие алгоритмы кластеризации

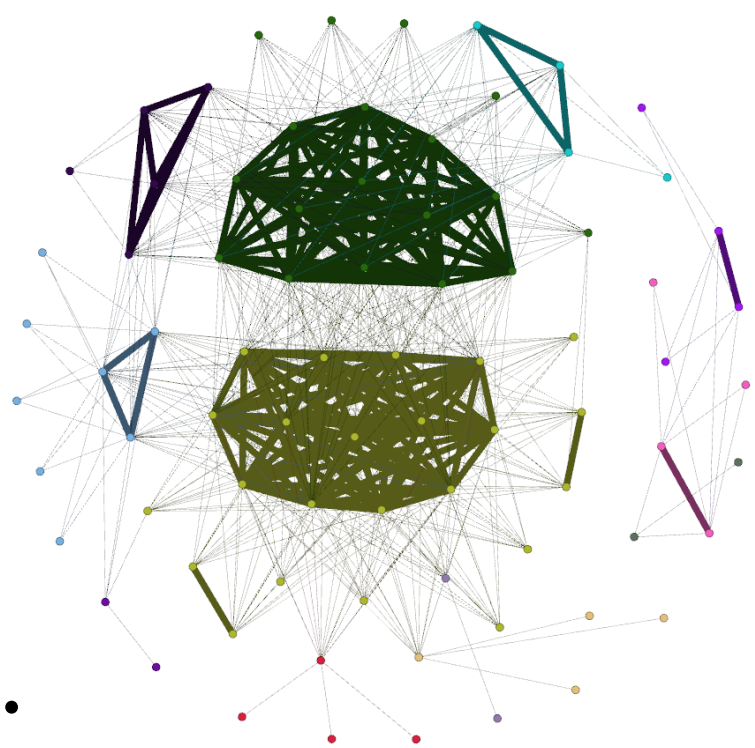
Вообще, задача кластеризации графа NP-полна. Однако существует множество приближенных алгоритмов. Их принято разделять на 3 основных типа:

1. Agglomerative. Изначально каждая вершина в своей компоненте, потом они как-то объединяются.
2. Divisive.Изначально одна большая компонента.
3. Hierarchical. Организация вершин в виде дерева, расстояние в котором между вершинами эквивалентно расстоянию в исходном графе.

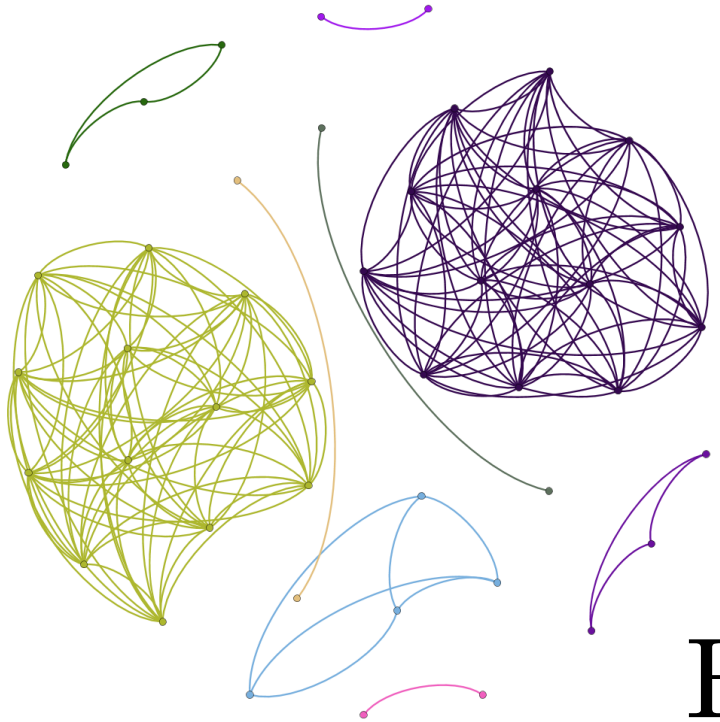
Наилучшие результаты дают алгоритмы, разбивающие граф на фиксированное количество частей.

Как найти количество комнонент?

Первая идея - это попытаться как-нибудь нарисовать графы и попробовать что-то увидеть.



A.



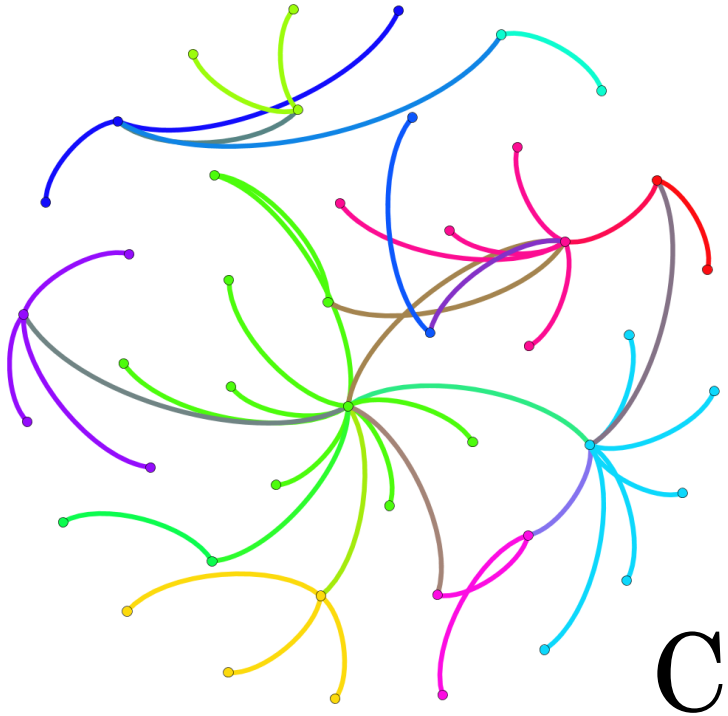
B.

Как найти количество комнонент?

Попробуем сжать граф по нулевым ребрам.

Идеи:

1. Просто нарисовать.
2. Нарисовать с единичными ребрами.
3. Нарисовать только с нулевыми ребрами.
4. Сжать граф по нулевым ребрам и нарисовать единичные.
5. Рисовать ребра с какой-то вероятностью.



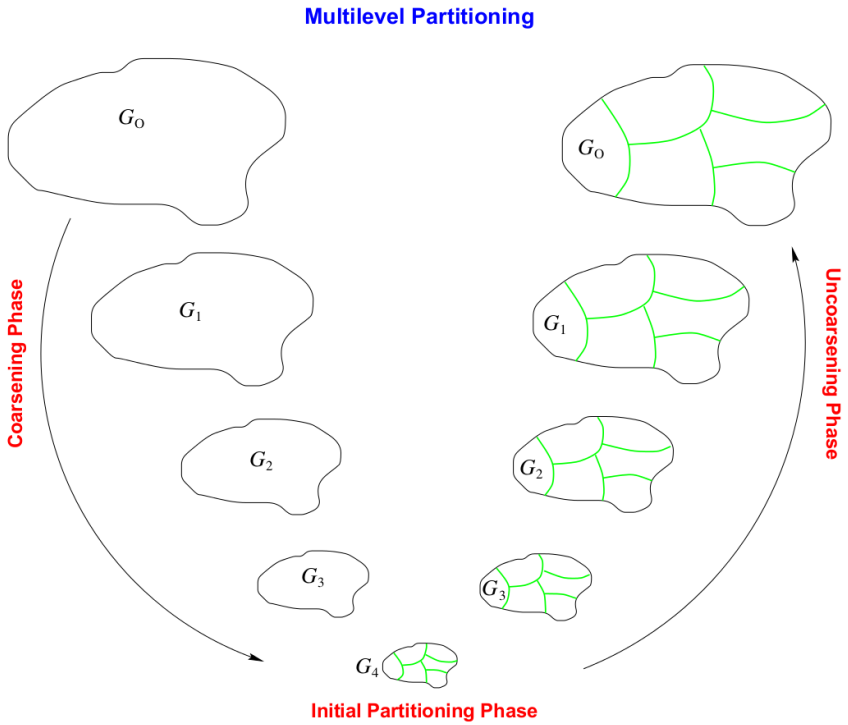
С.

Metis

A Software Package for Partitioning Unstructured Graphs…

Краткое описание алгоритма:

1. Будем сжимать исходный граф, пока количество вершин не будет достаточно мало.
2. Разобьем полученный граф на заданное количество частей используя алгоритм Кернигана-Лина.
3. Восстановление графа. Пока ответ можно улучшить, вершины мигрируют между компонентами.



Тестирование Metis-a

Нацелившись на metis стало интересно, насколько хорошо он будет решать нашу задачу. Получились слдующие результаты:

Среднее расстояние между строками базиса	Вероятность поломки одного символа	Успех
2 - eps	0.001	91%
2.5 - eps	0.001	99%
2 - eps	0.002	84%
2.5 - eps	0.002	95%

Только он не работает :(

Дальнейшие направления работы: Lloyd's algorithm и MCL

Lloyd's algorithm: выберем K точек (центров) минимизируя квадрат ошибки. Ошибка - вес пути между центром и всеми остальными точками.

Алгоритм:

- Пока центры меняются:
  - Пересчитаем расстояния и добавим каждую точку в кластер ближайшему центру.
  - В каждом кластере выберем новый центр.
- Ответ - центры с ближайшими точками.

MCL: coming soon