Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра Системного Программирования

Крамар Алина Сергеевна

Построение генетических карт по неполным и зашумленным данным

Дипломная работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
к. ф.-м. н., доцент Сысоев С. С.

Рецензент:
науч. сотрудник. Добрынин П.В.

Санкт-Петербург
2014

Saint Petersburg State University
Mathematics & Mechanics Faculty

Department of Software Engineering

Alina Kramar

Genetic linkage mapping based on incomplete and noisy data

Graduation Thesis

Admitted for defence.
Head of the chair:
professor Andrey Terekhov

Scientific supervisor:
associate professor Sergey Sysoev

Reviewer:
research associate Pavel Dobrynin

Saint-Petersburg
2014

Оглавление

Введение

Краткая медицинская энциклопедия [1] даёт следующее определение слову “генетика”: наука о наследственности и изменчивости организма. Согласно законам наследования все основные признаки и свойства любых организмов определяются и контролируются единицами наследственной информации - генами, локализованными в специфических структурах клетки - хромосомах [2, 3]. В связи с этим, основной задачей генетики является на основе первичной структуры биополимеров (молекул ДНК или РНК) определить фенотип особи, механизмы наследования тех или иных признаков и т.п. Получение информации о первичной структуре ДНК называется секвенированием [4]. К сожалению, современные методы секвенирования не в состоянии предоставить информацию о полной нуклеатидной последовательности в рамках конкретной хромосомы [5]. Сама по себе нуклеатидная последовательность не содержит прямой информации о происхождении того или иного однозначно идентифицируемого участка (маркера), и становится затруднительным определить, от какого предка был унаследован тот или иной признак. Для определения шаблонов наследования и выявления его принципов предназначены генетические карты хромосом [6].

Генетическая карта - это схема или порядок расположения маркеров на хромосоме (здесь и далее подразумевается структурный маркер, т.е. маркер, имеющий отличимое нуклеатидное представление, которое позволяет его идентифицировать) и генов. Зачастую, наличие генов не так важно, как наличие маркеров, потому что гены имеют свойство наследоваться не полностью [7]. Идея создать генные карты принадлежит Томасу Моргану, внёсшему неоспоримый вклад в теорию наследственности. Идея была основана на явлении сцепленного наследования генов. Из-за мейотического кроссинговера, который делает невозможным полностью скоррелированное наседование и влияет на расхождение сцепленных генов по разным гаметам, появилось предположение о связи физического расстояния между маркерами и их взаимодействии при наследовании [8]. Это предположение оправдалось, и в 1913 году ученик Моргана Альфред Стёртевант построил первую генетическую карту на основе данных Drosophila melanogaster [9, 10].

Кроссинговер на этапе мейоза вносит возмущение в сцепленное наследование [8], и чем чаще он проявляется, тем чаще наблюдается отклонение от сцеплений. Физическое расстояние между парой генов прямо пропорционально вероятности кроссинговера, что позволяет однозначно расположить маркеры на молекуле [10]. Генетическое расстояние не трудно перевести в физическое, но чаще всего в этом нет необходимости, так как для механизма наследования существенен порядок [11, 12]. Единицей измерения расстояния является 1 сантиморган. Стоит упомянуть, что задача построения генетической карты хромосомы имеет смысл для диплоидных особей и лучше решается, когда исследуемые особи находятся в родстве [13]. Задача так же решается

проще на семействах особей, которые размножаются с большой скоростью (больше данных), поэтому большее количество карт на данный момент имеют хромосомы дрозофил, кошек, мелких грызунов и насекомых [14]. Нас же интересуют генетические карты человека, так как эти карты являются единственным способом проведения генетического анализа на наличие и предрасположенность особи к тяжёлым наследственным заболеваниям. С помощью генетического анализа можно выявлять болезнь Альцгеймера [15], гемофилию [16], хорею Хаттингтона [17] и т.п.

Современные средства генетического картирования позволяют построить карту по форматированному файлу родословной [18, 19, 20]. Причём не обязательно иметь полную информацию о степени родства, половой принадлежности и возрасте. Более того, секвенирование допускает ошибки в порядке генов.

Общепризнанными [21] и наиболее распространёнными методами генетического картирования являются:

- Алгоритм Элстона-Стюарта
- Алгоритм Ландера-Грина

Программные средства, основанные на вышеуказанных алгоритмах хорошо решают поставленную задачу на взятом у человеческих особей материале при сравнительно небольшом (по сравнению с известными науке видами маркеров) количестве маркеров [Алгоритм Элстона—Стюарта] и для небольшого размера исследуемой семьи [Алгоритм Ландера—Грина]. Практические требования современной медицины приводят к необходимости строить генетические карты по все большим и большим наборам маркеров. Вычислительная сложность алгоритма Элстона-Стюарта растёт [22] экспоненциально по этому параметру, в результате чего генетическое картирование становится неосуществимым на практике. Алгоритм Ландера-Грина позволяет исследовать большее количество маркеров, но на меньших семействах, так как время работы этого алгоритма экспоненциально растёт с ростом количества наблюдаемых в родословной особей-“непрародителей” (особей, родители которых присутствуют в исследуемом множестве особей) [22].

В 2013 году в работе [23] был предложен алгоритм построения генетической карты путём прямого извлечения информации о генетических расстояниях между маркерами без учёта кратности кроссинговера. Вычислительная сложность данного алгоритма полиномиальна как от количества маркеров, так и от мощности родословной, что делает его перспективнее ранее упомянутых алгоритмов для решения задач современной генетики. При этом алгоритм прямого извлечения данных имеет ряд существенных недостатков, а именно:

1. Плохая теоретическая обоснованность

В работе [23] утверждается, что алгоритм извлекает всю возможную информацию о рекомбинациях в исследуемых особях, в главе **1.3** мы покажем, что это утверждение вообще говоря неверно.

2. Отсутствие верификации

   Алгоритм был проверен только на небольшой семье кошек из 192 особей и рассматривал 35 маркеров. На других данных алгоритм не проверялся.

3. Неверный результат в случае кратности кроссинговера,

   что существенно снижает полезность алгоритма, так как в природе неоднократный кроссинговер — явление, встречающееся достаточно часто.

Исходя из этого, цель данной работы можно сформулировать следующим образом: придумать новый алгоритм построения генетических карт на основе предложенного в статье [23] методе,

- усовершенствовав алгоритм в части учёта кратных рекомбинаций
- верифицируя полученный алгоритм на реальных и синтетических данных
- сравнив новый алгоритм с предшественниками

1. Обзор существующих решений

Основные понятия

Для того, чтобы рассматривать существующие подходы, нужно более формально поставить задачу, которую они решают. Нам потребуются следующие понятия [24]:

Маркер(ДНК-маркер) — полиморфный признак, выявляемый на уровне нуклеотидной последовательности ДНК.

Локус — положение маркера на генетической или цитологической карте.

Аллель — вариант последовательности ДНК в текущем локусе.

Гомозигота — диплоидная (двойной набор одинаковых хромосом) особь, копия генов которой представлена одинаковыми аллелями.

Гетерозигота — диплоидная особь, копия генов которой представлена разными аллелями.

Поясним введённые выше понятия примером. Информацию об особи будем записывать в виде строки вида **AaBb**, где **A** — вид маркера в отцовской хромосоме в позиции 1, **a** — вид маркера в материнской хромосоме в позиции 1, **B** — вид маркера в отцовской хромосоме в позиции 2, **b** — вид маркера в материнской хромосоме в позиции 2. В данном примере особь гетерозиготна в позициях 1 и 2, так как имеет разные маркеры от отца и матери. Если бы в позиции 2 находилась строка **BB**, то особь была бы гомозиготна, так как маркеры одинаковые.

Гамета — одинарный набор хромосом. В нашем случае, подстрока последовательности аллелей, содержащая половину генетического материала.

Фаза — различают две фазы: CIS и TRANS. CIS — расположение доминантных генов на одной хромосоме, а TRANS — расположение доминантных генов на разных хромосомах. Фаза имеет смысл только для особей, гетерозиготных в обоих локусах.

В нашем примере особь **AaBb** находится в фазе CIS, так как доминантные признаки (символы в верхнем регистре) находятся на одной хромосоме. **AabB** — пример TRANS фазы.

Рекомбинация — перераспределение генетического материала родителей в потомстве.

В приведённом выше примере при образовании особью гаметы, которая передастся потомкам, могут возникнуть следующие сочетания: **AB**, **Ab**, **aB** и **ab**. Получить информацию о наличии рекомбинации мы можем, зная фазу. Для CIS-фазы (в нашем случае, особь представляется строкой **AaBb** или **aAbB**), рекомбинантными будут являться гаметы **aB** и **Ab**, в случае же TRANS-фазы (особь — строка вида **AabB** или **aABb**), рекомбинантными являются гаметы **ab** и **AB**.

Генетическая карта — схема расположения маркеров на хромосоме.

Используя эти определения перейдём к алгоритмическому смыслу задачи и рассмотрим пути её решения. Имея на входе данные о родословной, количестве исследуемых маркеров, нуклеотидных последовательностях всех особей, а так же их родственной связи, генетическую карту можно построить следующими алгоритмами, имеющими программную реализацию [22]:

- Алгоритм Элстона — Стюарта
- Алгоритм Ландера — Грина
- Построение генетических карт по полностью секвенированным участкам геномов

1.1. Алгоритм Элстона-Стюарта

[25]
Описание - достоинства - недостатки

1.2. Алгоритм Ландера-Грина

[26, 27]
Описание - достоинства - недостатки

1.3. Построение генетических карт по полностью секвенированным участкам геномов

Описание из статьи - достоинства - недостатки

Построение генетических карт

Поскольку существующие алгоритмы имеют большую вычислительную сложность или не способны давать правильный ответ в наиболее частых случаях, было решено

написать новый алгоритм, позволяющий строить генетические карты быстрее и качественнее, чем у существующих. Так как в случае с экспоненциальным ростом (алгоритм Элстона-Стюарта и алгоритм Ландера-Грина) помочь могут только локальные оптимизации [28, 22, 29], не влияющие на асимптотическую скорость, разумно предположить, что развивать стоит алгоритм прямого извлечения данных о рекомбинациях из полностью секвенированных геномов, рассмотренный в главе **1.3**.

2. Постановка задачи

Обзор существующих решений показал, что ни один из существующих алгоритмов генетического картирования не является полноценным и удобным способом решения решения возникающих в современной генетике задач. Тем не менее недостатки алгоритма, предложенного в **1.3** могут быть устранены или сведены к минимуму.

Таким образом, целью данной работы является написание нового алгоритма генетического картирования на базе метода прямого извлечения данных.

Для достижения поставленной цели был сформулирован ряд задач:

1. Доработка алгоритма прямого извлечения данных
2. Верификация полученного алгоритма
   (a) моделирование и генерация тестовых данных
   (b) сравнение существующих алгоритмов с полученной версией

3. Доработка алгоритма

3.1. Анализ недостатков и их исправления

3.1.1. Восстановление расположения маркеров по матрице попарных расстояний

В статье [23] утверждается, что алгоритм использует всю информацию о происхождении аллели, которую возможно получить из предоставляемых данных. Для более точного представления процесса работы алгоритма была сформулирована его основная мысль: алгоритм упорядочивает набор маркеров на хромосоме, используя для этого матрицу попарных расстояний. Вообще говоря, непосредственно матрицу попарных расстояний на этапе построения мы получить не можем [30]. Поэтому алгоритм использует её оценку, основанную на частоте рекомбинаций. Как было сказано ранее, частота рекомбинаций между двумя маркерами прямо-пропорциональна расстоянию между ними [31], поэтому матрицу попарных расстояний мы оцениваем матрицей частот рекомбинации.

Так как реализация алгоритма не учитывает кратный кроссинговер, можно предположить, что оценка матрицы попарных расстояний наиболее точна в случае близлежащих маркеров, так как вероятность кроссинговера на отрезке малой длины меньше вероятности кроссинговера на отрезке большей длины. На Рис. 1 показан граф взаимных расстояний с вершинами. С помощью такого представления видно, что по полученным данным можно расположить маркеры в линию.

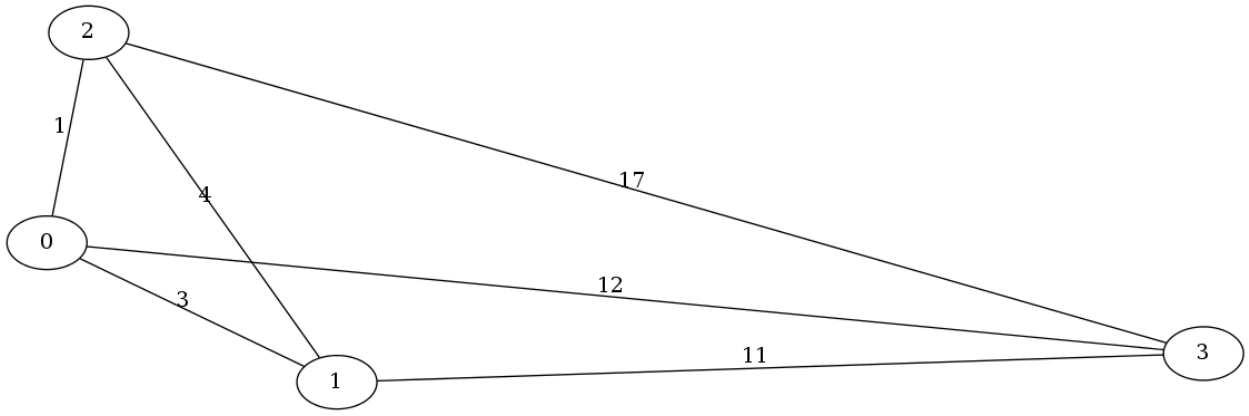

Рис. 1: Попарные расстояния в случае слабо удалённых маркеров

На Рис. 2 показана идентичная ситуация для выборки из четырёх маркеров, далеко расположенных друг от друга.

Таким образом мы показали, что матрица частот рекомбинации является оценкой матрицы попарных расстояний, причём оценка наиболее точна в точках, находящихся близко друг к другу.

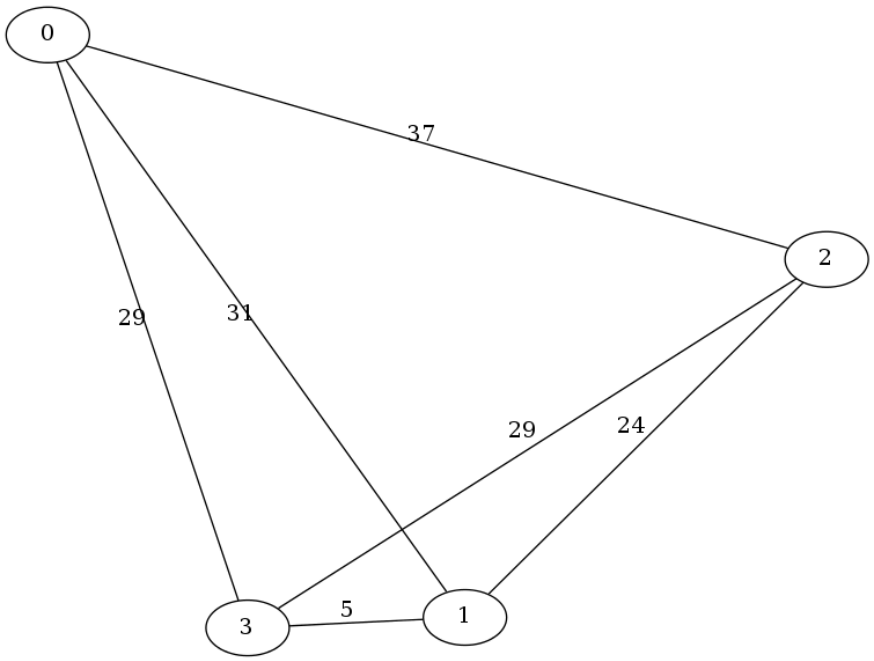

Рис. 2: Попарные расстояния в случае сильно удалённых маркеров

Исходя из приведённого выше замечания, мы считаем, что наиболее правдивыми данными в матрице являются пары маркеров с минимальными значениями частоты рекомбинаций между ними, поэтому начинаем строить карту от центра, представленного двумя самыми близкими маркерами. Более формально этот алгоритм можно записать следующим образом:

Алгоритм 1 Лианеризация маркеров

function LINMARK
  $n \leftarrow dim(distMatrix)$
  $(l, r) \leftarrow \text{argmin}_{x,y}\, distMatrix[x][y]$
  $order \leftarrow [l, r]$
  while есть не рассмотренные узлы **do**
    $l \leftarrow first(order)$
    $r \leftarrow last(order)$
    $m \leftarrow$ ближайший к l или r узел.
    if m ближе к r **then**
      $order \leftarrow order + [m]$
    else
      $order \leftarrow [m] + order$
    end if
  end while
  return order
end function

— где входным параметром $distMatrix$ является квадратная матрица частот рекомбинаций.

Применяя данный алгоритм, получаем результат в виде порядка маркеров. На Рис. 3 видно, хоть представление и не линейно, что есть некоторые ошибки, и что эти ошибки порядка допустимых транспозиций. Посмотрев на рисунок можно попробо-

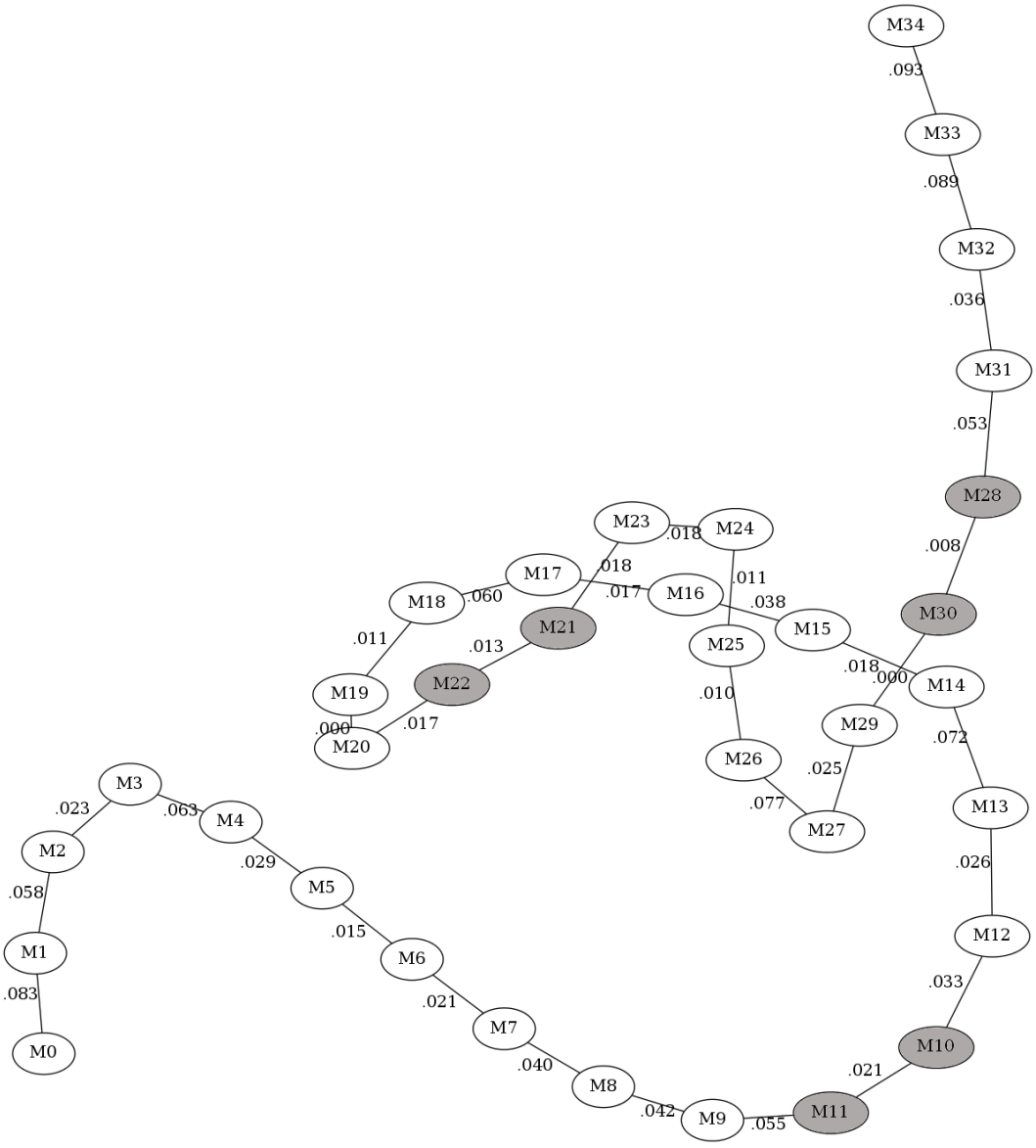

Рис. 3: Порядок маркеров после применения алгоритма вытягивания в линию на семействе из 192 кошек с 35 маркерами

вать представить изначальную матрицу расстояний в виде графа с вершинами в маркерах и рёбрами с весами, равными частоте рекомбинаций между двумя вершинами (в нашем случае — маркерами). Если вспомнить алгоритм построения минимального остовного дерева Прима [32], который основывается на выборе ребра с минимальным весом, что в нашем случае эквивалентно выбору пары маркеров с минимальным расстоянием между ними среди всех остальных пар, то логичным является попробовать находить порядок маркеров с помощью алгоритма Прима. Заметим так же, что в случае правильно построенной матрицы попарных расстояний, минимальным основ-

ным деревом будет являться искомый порядок, который превратится в линию. На

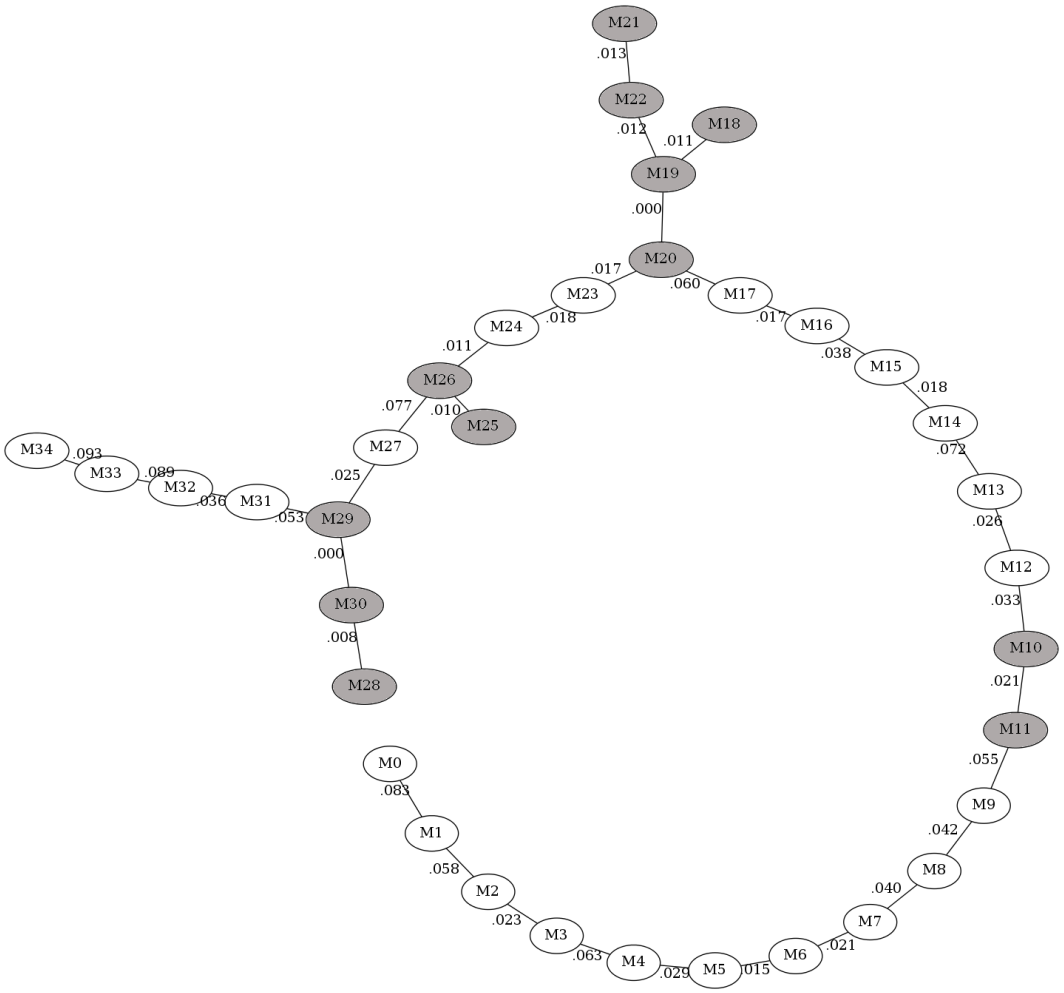

Рис. 4: Минимальное остовное дерево графа попарных расстояний

Рис. 4 представлено построенное с помощью алгоритма Прима минимальное остовное дерево. Не трудно заметить, что те участки, что не имели нарущения порядка в первом варианте остались неизменными, но появились особенности, которые не дают полученному графу называться линией и упорядочить вершины. Это особенности, вызванные либо парами, между которыми не наблюдалось рекомбинаций (.000 на графе), либо парами маркеров, лежащих друг от друга на большом расстоянии.

Данный пример показал, что в таких задачах применимы основы теории графов и предположения о наибольшем правдоподобии маленьких расстояний. Кроме того, показанный алгоритм при правильной организации хранения графа имеет вычислительную сложность $O(M^2 * log(M))$, что вносит небольшое улучшение в алгоритм, так как в изначальной версии алгоритма сложность вытягивания матрицы в линию составляла $O(M^3)$. Воспользоваться алгоритмом Прима мы сможем в случае, когда матрица частот рекомбинаций наиболее точно приближает матрицу попарных расстояний.

3.1.2. Повышение точности матрицы частот рекомбинаций

Рассмотренные выше локальные оптимизации базируются на том, что матрица попарных расстояний, полученная в результате извлечения информации о происхождении аллели правдоподобна. Мы уже показывали, что для далеко отстающих друг от друга маркеров оценка не верна.

Не трудно заметить, что точность получаемого результата зависит от точности матрицы частот рекомбинаций, так как в нашем случае, именно эта матрица является приближением матрицы попарных расстояний. Как было неоднократно замечено выше, алгоритм, на базе которого мы пытаемся получить новое решение, не учитывает кратность кроссинговера, в связи с чем, матрица частот рекомбинации актуальна только в случае однократного кроссинговера. Резонной является задача извлечения данных о кратных рекомбинациях, так как это явление случается в природе достаточно часто.

Давайте попробуем рассмотреть функцию от расстояния между маркерами $f(d) : \mathbb{R} \to \mathbb{N}$, возвращающую количество произошедших рекомбинаций. Про эту функцию мы можем смело сказать, что она неубывающая, потому что если рекомбинация произошла на участке, разделяющем два маркера, то на более большом участке рекомбинаций возможно не меньше, чем уже произошли. Однако первоначально нам не известен порядок маркеров и расстояния между ними.

Заметим, что частичные сведения о порядке и расстояниях нам даёт один проход алгоритма по родословной. Естественным образом встаёт вопрос о возможностях уточнения полученного порядка.

Нами было выдвинуто предположение, что базируясь на полученном после одного прохода результе, можно уточнить его рассматривая каждую особь повторно, аккумулируя матрицу частот рекомбинаций, зная примерное расстояние между маркерами.

3.2. Доработанный алгоритм

Так как нахождение матрицы частот рекомбинаций является самым главным этапом алгоритма генетического картирования, а точность полученный матрицы влияет на правильность результата работы всего алгоритма, то основную задачу данной работы можно сформулировать следующим образом: найти алгоритм уточнения матрицы частот рекомбинаций.

Нами было учтено предположение, что можно использовать результат работы изначальной версии алгоритма как предположение о порядке расположении маркеров на хромосоме. При данном допущении мы можем использовать знания о количестве произошедших рекомбинаций в зависимости от расстояния между ними.

Исходя из этого, получаем дополненный алгоритм, основанный на изначальной версии, входные параметры которого аналогичны с рассмотренным в главе **1.3**:

Алгоритм 2 Построение генетических карт с учётом кратности кроссинговера

```
1: function GENMAP2
2:     order ← gen_map(pedegree)
3:     agg ← 0_{M,M}
4:     for k in pedegree do
5:         recs ← 0_{M,M}
6:         for i ← 0; i < M; i++ do
7:             for j ← i; j < M; j++ do
8:                 recs[i, j] ← getRecCount(k, i, j)
9:                 recs[j, i] ← getRecCount(k, i, j)
10:            end for
11:        end for
12:        for i ← 0; i < M − 1; i++ do
13:            for j ← 0; i < M − 1; j++ do
14:                if recs[i, j] > recs[i, j + 1] and recs[i, j + 1] = 0 then
15:                    recs[i, j + 1] ← recs[i, j] + 1
16:                    recs[j + 1, i] ← recs[j, i] + 1
17:                end if
18:                if recs[i, j] > recs[i, j + 1] and recs[i, j + 1] = −1 then
19:                    recs[i, j + 1] ← recs[i, j]
20:                    recs[j + 1, i] ← recs[j, i]
21:                end if
22:            end for
23:        end for
24:        agg ← agg + recs
25:    end for
26:    return Linmark(agg)
27: end function
```

Новый алгоритм алгоритм реализует описанную выше схему. Покажем, что эта схема оптимальная и более быстрая, чем наиболее популярные сейчас алгоритмы. А так же обоснуем преимущества данной версии алгоритма по вравнению с предыдущей реализацией.

4. Сравнение алгоритмов

Существует два подхода к сравнению биологических алгоритмов:

- сравнение с использованием реальных данных.
- сравнение с использованием синтетических данных.

Зачастую провести корректное сравнение при помощи реальных данных не удаётся по следующим причинам:

- Биологические данных не много.

  Секвенирование молекул ДНК не производится для некоторых видов по причине дороговизны или отсутствия научного интереса к этим данным.

- Данные из разных источников могут быть не сопостовимы

  Например, достаточно мало данных, которые отличаются минимальным набором параметров. Трудно найти секвенированные родословные 250 собак, если в одном случае нужны 250 особей только белого цвета, а в другой 250 собак, отличающихся цветом глаз. Чаще всего встречаются разрозненные наборы данных, с разными количеством и видами маркеров, количеством поколений и т.п.

- Для биологических данных характерны ошибки.

  Хоть и утверждается, что вероятность ошибок при севенировании минимальна, оги все-таки случаются. Получение природного материала не может быть однозначным по определению, так как велик человеческий фактор и недостатки биологического программного обеспечения.

- Получение биологических данных дорого.

  Секвенирование особей и тем более целых родословных стоит дорого. Например, полное секвенирование генома человека стоит не менее 1000 USD [33]. Для целей тестирования и отдадки методов это несоизмеримые затраты.

Синтетические данные лишены описанных выше недостатоков,но при неправильной модели синтеза данных они могут и не соответсвовать действительности, искажать или вовсе потерять биологический смысл.

Именно поэтому для сравнения алгоритмов между собой, а так же для верификации новой версии алгоритма нами была выбрана следующая стратегия: сравнить на синтетике, чтобы обнаружить сильные и слабые стороны алгоритмов и затем верефицировать на сраных кошках.

Сравнение существующих алгоритмов на синтетических данных является важной задачей, так как это позволяет выявить немало полезной информации, такой как наиболее подходящие случаи для использования, вероятные ошибки, вычислительную сложность. Кроме того, данные можно генерировать дл акцентирования внимания на исследуемой проблеме. Например, в случае исследования поведения программы на данных, полученных путём моделирования кратного кроссинговера. Заметим, что подобную информацию трудно получить из естественных результатах секвенирования. Кроме того, параметризация генерируемых данных упрощает проверку результата работы алгоритма.

4.1. Генерация тестовых данных

Генерация тестовых данных в случае генетики является задачей, требующей аккуратности, так как во время моделирования сгенерированные тестовые данные могут потерять биологический смысл. При решении этой задачи нами было выявлено 2 принципиально разных подхода:

- генерирование данных по заданным параметрам
- генерирование данных на основе известного результата

В первом случае, генерируя тестовое множество особей, нам необходимо понимать, каким образом выглядит итоговая и желаемая генетическая карта.Во втором, имея карту, легко сравнивать получаемые результаты работы алгоритма. Этот подход заведомо сложнее, так как никто не гарантирует биекцию между множествами входных данных и множеством результатов работы алгоритма на этих входных данных. В связи с этим мы выбрали первый подход.

Входные параметры:

- количество особей-прародителей (N)
- количество прямых потомков (детей) (F1)
- количество поколений (gen_count)
- итоговое количество особей в родословной (common_N)
- вероятность рекомбинации при мейозе (rec_prob)
- количество рассматриваемых маркеров (markers_count)

Особь в нашем случае, это объект класса Organism
// тут будет код и его описание в виде листинга

Алгоритм 3 Генерация родословной

```
 1: procedure GenPedigree
 2:     population ← []
 3:     founders ← список прародителей
 4:     population ← population + founders
 5:     for gen ← 0; gen < n_gen; gen++ do
 6:         (males, females) ← divide(population)
 7:         mother ← choice(females)
 8:         father ← choice(males)
 9:         child ← CrossBreed(mother, father)
10:         добавить child к population
11:     end for
12:     print population
13: end procedure
14:
15: function CrossBreed(mother, father)
16:     mothergamet ← Crossover(motherchromosomes)
17:     fathergamet ← Crossover(fatherchromosomes)
18: end function
19:
20: function Crossover(chromosomeA, chromosomeB)
21:     crossovers ← точки кроссинговера
22:     gametA ← []
23:     gametB ← []
24:     for position in chromosome do
25:         if position in crossovers then
26:             swap chromosomeA, chromosomeB
27:             gametA ← gametA + chromosomeA[position]
28:             gametB ← gametB + chromosomeB[position]
29:         end if
30:     end for
31:     return random(gametA, gametB)
32: end function
```

4.2. Процесс сравнения и анализа

Для наиболее хорошего и полного сравнения алгоритмов стоит выделить ряд биологических аспектов, влияющих на результаты работы алгоритмов:

- наличие большого количества неинформативных пар (много гомозиготных локусов)
- наличие неоднократного кроссинговера
- его чётность

Не стоит забывать, что немаловажным для скорости выполнения алгоритма является порядок количества маркеров и мощность родословной.

В связи с этим рассмотрим сочетания биологических особенностей, влияющих на качество результатов алгоритмов, с объёмом данных.

ТАБЛИЧКА

4.3. Обобщённые данные

организмы and маркеры	30	40	50
200	o200m30	o200m40	o200m50
300	o300m30	o300m40	o300m50
400	o400m30	o400m40	o400m50

Таблица 1: Входные даные

Другие данные:

nocross 200 кошек 30 маркеров без кросинговеров

singlecross 200 кошек 30 маркеров без кратных кроссинговеров

manycross 200 кошек 30 маркеров и очень много кроссинговеров

cats 192 кошки блеать

	сыс	expO	expm
nocross	0	0	0
singlecross	0	0	0
manycross	10	2	3
cats	1	0	0

Таблица 2: Количество ошибок

Запуск на этих данных показал, что алгоритм CCC работает намного быстрее двух других, что его качество уменьшается с увеличением эффекта от кратных кроссинговеров, и что для реальных биоданных его точности более чем хватает.

	сыс	expO	expm
o200m30	1c	10m	1h
o200m40	1c	10m	2d
o200m50	2c	10m	∞
o300m30	2c	1d	2h
o300m40	3c	1d	2d
o300m50	3c	1d	∞
o400m30	5c	∞	3h
o400m40	6c	∞	3d
o400m50	8c	∞	∞

Таблица 3: Скорость работы алгоритмов

Заключение

Результаты

В ходе выполнения дипломной работы нами был предложен доработанный алгоритм прямого извлечения информации о рекомбинациях из секвенированного генома, позволяющий получать генетические карты в случае одно- и многократного кроссинговера. Для верификации нового алгоритма был реализован механизм тестирования методов построения генетических карт, с помощью которого были наглядно представлены преимущества нашей реализации алгоритма.

Актуальность полученных результатов

Применение генетических карт получило достаточно широкое распространение в сфере диагностирования наследственных заболеваний. Исследование механизмов наследования, а так же выявление его закономерностей позволяет получить информацию о предрасположенности у человека (и не только) к тем или иным отклонениям, которые трудно выявить до начала проявления симптомов. Получение информации о генной предрасположенности до начала заболевания позволяет моделировать лечения и образ жизни, избегая или купируя их дальнейшие проявления.

Кроме того, генетические карты применяются в исследованиях процессов эволюции. Рассматривая генетические карты двух достаточно близких видов, можно извлечь информацию о том, в каком направлении эволюционировала фауна и какие гены передавались в первую очередь.

Из-за недостатков современных методов секвенирования, а так же из-за отсутствия возможности секвенировать гаметы, единственными способами построить генетическую карту являются рассмотренные выше алгоритмы. Предложенный нами алгоритм позволяет строить карты быстрее, чем альтернативные алгоритмы, и точнее, чем его предшественник.

Список литературы

[1] Б.В. Петровский. Краткая медицинская энциклопедия. *Сов. Энциклопедия*, 1989.

[2] Anthony JF Griffiths. *An introduction to genetic analysis*. Macmillan, 2005.

[3] Anthony JF Griffiths, Jeffrey H Miller, David T Suzuki, and Richard C Lewontin. An introduction to genetic analysis. 2000.

[4] Б Албертс, Д Брей, Дж Льюис, К Роберте, and Дж Уотсон. Молекулярная биология клетки. В трех томах, 1994.

[5] Strachan T. and Read E. *Human Molecular Genetics*. Garland Science, New York, 4 edition, 2009.

[6] Thomas Hunt Morgan, Alfred Henry Sturtevant, Hermann Joseph Muller, and Calvin Blackman Bridges. *The mechanism of Mendelian heredity*. Holt, New York, 1922.

[7] Jon Schiller. *Genome Mapping: To Determine Disease Susceptibility*. CreateSpace, 2010.

[8] Harriet B Creighton and Barbara McClintock. A correlation of cytological and genetical crossing-over in zea mays. *Proceedings of the National Academy of Sciences of the United States of America*, 17(8):492, 1931.

[9] Susan Goldhor. Genetics and evolution—selected papers of ah sturtevant. *The Yale journal of biology and medicine*, 34(5):528, 1962.

[10] Alfred Henry Sturtevant, George Wells Beadle, et al. An introduction to genetics. *An introduction to genetics.*, 1939.

[11] Daniel Hartl and Maryellen Ruvolo. *Genetics*. Jones & Bartlett Publishers, 2011.

[12] George M Malacinski. *Essentials of molecular biology*. Jones & Bartlett Learning, 2005.

[13] В. В. Хвостова and Ю. Ф. Богданов. *Цитология и генетика мейоза*. Наука, Москва, 1975.

[14] MARY SARA McPEEK. An introduction to recombination and linkage analysis. In *Genetic Mapping and DNA Sequencing*, pages 1–14. Springer, 1996.

[15] Gerard D Schellenberg, Thomas D Bird, Ellen M Wijsman, Harry T Orr, Leojean Anderson, Ellen Nemens, June A White, Lori Bonnycastle, James L Weber, M Elisa Alonso, et al. Genetic linkage evidence for a familial alzheimer's disease locus on chromosome 14. *Science*, 258(5082):668–671, 1992.

[16] Isabelle Oberle, Giovanna Camerino, Roland Heilig, Lelia Grunebaum, Jean-Pierre Cazenave, Calogero Crapanzano, Pier M Mannucci, and Jean-Louis Mandel. Genetic screening for hemophilia a (classic hemophilia) with a polymorphic dna probe. *New England Journal of Medicine*, 312(11):682–686, 1985.

[17] MARY ANNE ANDERSON, RUDOLPH E TANZI, PAUL C WATKINS, KATHLEEN OTTINA, MARGARET R WALLACE, ALAN Y SAKAGUCHI, ANNE B YOUNGH, IRA SHOULSONH, and ERNESTO BONILLAH. *A polymorphic DNA marker genetically linked to Huntington's disease*, volume 51. Oxford University Press, 2004.

[18] Eric S Lander, Philip Green, Jeff Abrahamson, Aaron Barlow, Mark J Daly, Stephen E Lincoln, and Lee Newburg. Mapmaker: an interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics*, 1(2):174–181, 1987.

[19] Phil Green, Kathy Falls, and Steve Crooks. *Documentation for CRI-MAP*, 2.4 edition, 1990.

[20] Elizabeth Thompson. *Introduction to lm_ map.*, 2010.

[21] Leonid Kruglyak, Mark J Daly, Mary Pat Reeve-Daly, and Eric S Lander. Parametric and nonparametric linkage analysis: a unified multipoint approach. *American journal of human genetics*, 58(6):1347, 1996.

[22] Maáyan Fishelson and Dan Geiger. Exact genetic linkage computations for general pedigrees. *Bioinformatics*, 18(suppl 1):S189–S198, 2002.

[23] Сысоев С.С. *Эффективный алгоритм построение генетических карт по полностью секвенированным участкам геномов.* Издательство С.-Петербургского Университета, СПб, 2013.

[24] *Биологический энциклопедический словарь.* Советская Энциклопедия, Москва, 1986.

[25] Robert C Elston and John Stewart. A general model for the genetic analysis of pedigree data. *Human heredity*, 21(6):523–542, 1971.

[26] Eric S Lander and Philip Green. Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences*, 84(8):2363–2367, 1987.

[27] E.S. Lander and P. Green. *The Progress of Genetic Research*, volume 84 (8). National Academy of Sciences, USA, 1987.

[28] Jeffrey R O'Connell. Rapid multipoint linkage analysis via inheritance vectors in the elston-stewart algorithm. *Human heredity*, 51(4):226–240, 2001.

[29] Maáyan Fishelson and Dan Geiger. Optimizing exact genetic linkage computations. *Journal of Computational Biology*, 11(2-3):263–275, 2004.

[30] AJ Bohonak. Ibd (isolation by distance): a program for analyses of isolation by distance. *Journal of Heredity*, 93(2):153–154, 2002.

[31] Piet Stam. Construction of integrated genetic linkage maps by means of a new computer package: Join map. *The Plant Journal*, 3(5):739–744, 1993.

[32] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, et al. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.

[33] Martin Kircher and Janet Kelso. High-throughput dna sequencing–concepts and limitations. *Bioessays*, 32(6):524–536, 2010.