САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

Вероятностные методы планирования движения и их применение к анализу поведения протеинов

Дипломная работа студента 545 группы

Катышева Алексея Александровича

Научный руководитель	…… / подпись /	к.ф.-м.н. Вяткина К. В.
Рецензент	…… / подпись /	д.ф.-м.н., проф. Степанов Е. О.
“Допустить к защите” заведующий кафедрой	…… / подпись /	д.ф.-м.н., проф. Терехов А. Н.

Санкт-Петербург

2011

SAINT PETERSBURG STATE UNIVERSITY

Mathematics and Mechanics Faculty

Software Engineering Department

Probabilistic methods of motion planning and their applications to protein behavior analysis

Graduate paper by

Katyshev Alexey

Scientific advisor	…… / signature /	Ph.D. K. V. Vyatkina
Reviewer	…… / signature /	Ph.D., Professor E. O. Stepanov
"Approved by" Head of Department	…… / signature /	Ph.D., Professor A. N. Terekhov

Saint-Petersburg

2011

Содержание

Введение 4
Анализ движения протеинов 4
Планирование движения (Motion Planning) 6
1. Постановка задачи 8
2. Обзор существующих решений 9
2.1. Молекулярная динамика 9
2.2. Симуляция Монте-Карло 9
2.3. Приближенные методы поиска траектории 10
2.4. Стохастическая карта дорог 11
3. Описание реализуемого подхода 12
3.1. Вероятностная карта дорог 12
3.2. Энергетическая функция 12
3.3. Генерация вершин графа 14
3.4. Соединение вершин графа рёбрами 14
3.5. Ответ на запрос 16
4. Особенности реализации 17
4.1. Входные и выходные данные 17
4.2. Карта дорог 18
4.3. Пространственная геометрия 20
5. Полученные результаты 24
6. Заключение 27
Список литературы 28

Введение

Анализ движения протеинов

Протеины (белки) – высокомолекулярные органические вещества, состоящие из соединённых в цепочку пептидной связью аминокислот. У живых организмов аминокислотный состав белков определяется на генетическом уровне.

Фредерик Сенгер в 1954 году с помощью разработанного им же метода секвенирования белков определил аминокислотную последовательность первого белка – инсулина. В 1958 году Максом Перуцом (Max Perutz) и Джоном Кендрю (John Kedrew) при помощи метода дифракции лучей были получены первые данные о трёхмерных структурах белков гемоглобина и миоглобина, соответственно. В 1971 году Уолтер Хэмилтон (Walter Hamilton) создал банк данных 3D структур белков и нуклеиновых кислот, названный впоследствии Protein Data Bank. С тех пор определены пептидные цепи и трёхмерные структуры для многих белков, и на данный момент Protein Data Bank насчитывает около 75000 структур.

Белки вовлечены в разнообразные процессы организма, такие как движение мышц, транспортировка веществ между клетками, распознавание и уничтожение антител, катализация множества важных химических реакций. Однако для того, чтобы выполнять свои функции, после синтеза каждый белок должен принять определенную трехмерную структуру, называемую иногда нативным состоянием. Каждая аминокислота транспортируется в клетку молекулой РНК. В клетке аминокислоты соединяются друг с другом пептидной связью, чтобы сформировать последовательность аминокислотных остатков, называемую полипептидной цепью. Эта цепь затем и проходит упомянутый выше процесс преобразования, при котором формируется уникальная трехмерная структура, которая компактна и устойчива. Этот процесс принято называть сворачиванием белка (protein folding). Полипептидная цепь может принимать невероятное число

различных форм. Если просто предположить, что каждый аминокислотный остаток может иметь 4 дискретных состояния (что не совсем верно), то даже самые короткие цепи (в которых обычно не меньше 50 аминокислотных остатков) могут принимать до 4^{50} состояний. Даже если не принимать в рассмотрение невозможные состояния (цепь не может иметь самопересечений, и никакие 2 атома не могут быть слишком близко друг к другу), то все равно пространство возможных состояний остается очень велико. Сворачивание же некоторых белков происходит за десятые доли миллисекунд. Поэтому очевидно, что сворачивание – определенно не случайный процесс.

На данный момент не очень много известно о процессах сворачивания белка. Понимание этих процессов помогло бы определить, почему некоторые белки не сворачиваются должным образом к своему нативному состоянию, то есть происходит так называемой мисфолдинг белка (protein misfolding). На данный момент считается, что именно мисфолдинг приводит к таким заболеваниям как болезнь Альцгеймера, болезнь Хантингтона и многим другим. Есть вероятность, что, изучив процессы сворачивания белков, учёные найдут способ блокировать преобразования, ведущие к мисфолдингу.

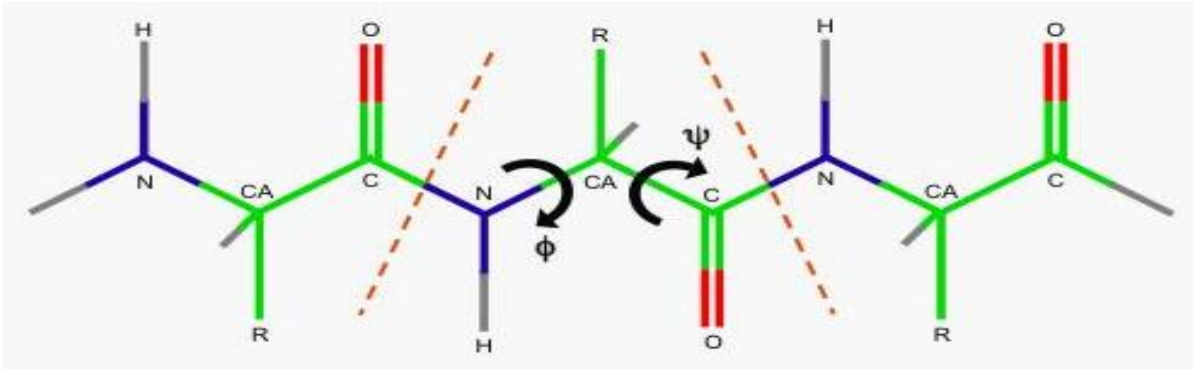

Рис. 1 Часть полипептидной цепи

Рассмотрим геометрическую модель белка. Часть полипептидной цепи представлена на Рис. 1 [1]. Пунктирными линиями выделен отдельный аминокислотный остаток, буквой R обозначена замещающая группа (своя для каждого из 20 видов аминокислот). Известно, что основными степенями свободы каждого белка являются торсионные углы (двугранные углы между

определенными полуплоскостями), отмеченные на рисунке буквами φ и ψ. Остальные величины (расстояния между атомами, плоские и прочие двугранные углы) в рамках одной полипептидной цепи могут считаться константами. Поэтому для цепи, состоящей из N+1 аминокислотного остатка, имеется 2N степеней свободы (для каждого из крайних остатков один из углов не определен). Таким образом, полипептидная цепь может иметь сотни и даже тысячи степеней свободы. Для моделирования поведения системы такой сложности хорошо подходят некоторые методы планирования движения.

Планирование движения (Motion Planning)

Планирование движения – изначально раздел робототехники, включающий в себя разнообразные методы и подходы, основной задачей которых является нахождение эффективного перехода системы из одного состояния в другое. В качестве примера можно привести задачу по перемещению механического робота в комнате с препятствиями из одной точки комнаты в другую. Также методы планирования движения нашли применение и в других областях, таких как анимация, архитектурное проектирование, молекулярная биология и др.

Чаще всего задача планирования движения состоит в том, чтобы определить последовательность допустимых и, желательно, наиболее эффективных (в том или ином смысле) промежуточных состояний и преобразовать данное начальное состояние в некоторое желаемое конечное состояние.

Практически все методы работают с так называемым конфигурационным пространством (configuration space, C-space). В этом пространстве любое состояние системы представляется точкой (обычно размерность этого пространства равно количеству степеней свободы системы). Тем самым задача сводится к поиску кратчайшего пути из одной

точки в другую в многомерном пространстве. Но при этом возникает две основные сложности:

- в конфигурационном пространстве есть области недопустимых состояний
- метрики в этом пространстве могут быть самыми разнообразными

В рамках данной работы для реализации был выбран один из основных методов планирования движения - построение вероятностной карты дорог (далее Probabilistic Roadmap, или просто PRM) в конфигурационном пространстве. Карта дорог – это ориентированный граф с весами на рёбрах, причём вершины этого графа случайным образом генерируются в пространстве (причем точки в недопустимых областях отсеиваются), а рёбра полностью лежат в допустимых областях, причем вес каждого рёбра – это стоимость перехода из одного состояния в другое.

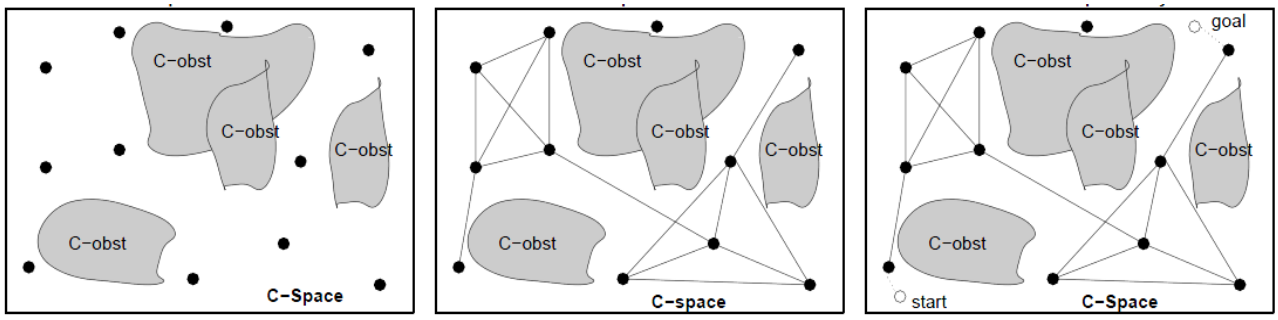

Рис. 2 Построение PRM (a-b) и ответ на запрос (c)

На Рис. 2 [2] проиллюстрированы основные этапы работы алгоритма:

- (a) – генерация вершин графа
- (b) – соединение вершин рёбрами друг с другом
- (c) – использование графа для ответа на запрос

После построения графа процесс ответа на запрос вида «из состояния A в состояние B» очень сильно упрощается. Подробнее об этом и о подходе в целом см. в разделе 3

1. Постановка задачи

Цель данной работы – реализовать алгоритм построения PRM для решения конкретной задачи преобразования пептидной цепи протеинов, проанализировать результаты работы алгоритма на реальных данных и сделать выводы о применимости полученной реализации.

Для достижения поставленной цели были выделены следующие этапы работы:

1. Разработка библиотеки, содержащей в себе реализацию алгоритма построения Probabilistic Roadmap, а также интерфейсы настройки под задачи различных типов.
2. Разработка библиотеки для работы с протеинами, в том числе чтение и запись файлов формата .pdb (Protein Data Bank)
3. Настройка Probabilistic Roadmap под конкретную задачу
4. Проведение экспериментов на реальных данных
5. Анализ полученных результатов, сравнение с другими реализациями

2. Обзор существующих решений

2.1. Молекулярная динамика

Симуляция при помощи законов молекулярной динамики - очень трудоёмкая и точная модель, чаще всего применяемая для быстро сворачивающихся белков, моделирования колебаний стабильного состояния и циклических движений. Такая модель, используя законы физики, по исходному состоянию моделирует состояние протеина через некоторый интервал времени dt. Как правило, в качестве этой величины берется одна десятая периода наиболее скоротечного движения системы, которым чаще всего оказывается колебание связи C-H. Эта связь колеблется с периодом в 10 фемтосекунд (одна фемтосекунда = 10^{-15} секунды), тем самым dt = 1 фс. Но большинство сложных протеинов сворачиваются минимум за десятые доли миллисекунд, что означает, что для моделирования процесса сворачивания потребуется порядка 10^{11} шагов.[3][4]

В 2000 году учёными из Стэнфордского университета (Stanford University) запущен проект распределённых вычислений для проведения компьютерной симуляции свёртывания белковых молекул, названный Folding@home[5]. Вычисления основаны на вышеописанном подходе. К настоящему моменту успешно смоделирован процесс свёртывания большинства белковых молекул продолжительностью 5-10 мкс. В данный момент производительность проекта составляет примерно 5,6 петафлопс.

2.2. Симуляция Монте-Карло

Симуляция Монте-Карло - это распространенный способ изучения термодинамических свойств молекулярных систем.

Симуляция начинается с некоторого начального состояния, затем строится случайный путь в сопровождающем пространстве. Каждая следующая точка пути выбирается случайным образом вокруг текущей (обычно по нормальному распределению), и переход осуществляется с

некоторой вероятностью, которая зависит от потенциальной энергии системы в этих точках. Процесс останавливается либо после некоторого фиксированного количества шагов, либо при достижении стабильного состояния.[6]

Оба приведенных выше метода выдают очень хороший результат, но при этом имеют два основных недостатка:

- за один раз строится лишь одна траектория. В данный момент считается, что сворачивание белка происходит в некотором энергетическом туннеле в сопровождающем пространстве, и поэтому очень важно знать множество путей перехода
- методы очень зависимы от локальных минимумов энергетической функции. Обычно энергетическая функция имеет много областей локальных минимумов, и приведенные методы, попав в эти области, тщетно пытаются «сбежать» оттуда.

2.3. Приближенные методы поиска траектории

Существует множество приближенных методов поиска путей сворачивания из одной конкретной конфигурации в другую.

Например, методы Self Penalty Walk [7] и Nudged Elastic Band [8] находят пути между двумя локальными минимумами энергетической функции. Сначала строится приближенная дискретная траектория (обычно при помощи линейной интерполяции). Затем посредством изменения промежуточных конфигураций эта траектория перестраивается. Промежуточные конфигурации изменяются посредством минимизации метаэнергетической функции. Эта функция учитывает все промежуточные конфигурации траектории, а также взаимодействия между ними. Метод SPW был успешно применен для поиска пути между двумя конфигурациями миоглобина.

Метод стохастического разностного уравнения (Stochastic Difference Equation [9]) заключается в решении краевой задачи и нахождении

устойчивой точки для функции работы $Y = \int_A^B \sqrt{2(E-U)}dl$, где A и B – начальная и конечная конфигурации, E – общая энергия, U – потенциальная энергия, dl – элемент длины. В [9] описывается применение данного метода к поиску траекторий свёртывания протеина А.

Вышеописанные подходы, также как и ранее описанные методы, имеют один существенный недостаток – их результатом являются конкретные траектории.

2.4. Стохастическая карта дорог

Основной целью этого метода является приближенное вычисление вероятности сворачивания протеина к конкретной конфигурации. Основной работой по этой теме является [10]. Стохастическая карта дорог является усовершенствованием вероятностной карты дорог (подробнее о которой далее, см. 3). В данном подходе вес каждого ребра представляет собой вероятность перехода из одного состояния в другое, а сам граф рассматривается как Марковская цепь. Также, в отличие от приведенных в пунктах 2.1 и 2.2 методах, стохастическая карта дорог позволяет обрабатывать очень большое множество путей одновременно. Это позволило добиться точности, сравнимой с точность метода Монте-Карло, при уменьшении времени работы алгоритма на несколько порядков.

3. Описание реализуемого подхода

3.1. Вероятностная карта дорог

Метод построения вероятностной карты дорог был предложен группой ученых из Стэнфордского Университета и Университета Утрехта [11]. Изначально он был предназначен для моделирования передвижений робота в статичном окружении. Весь метод можно разделить на две основные стадии: стадия обучения и стадия выполнения запросов. Под стадией обучения здесь понимается построение карты дорог в конфигурационном пространстве, под стадией выполнения запросов – поиск при помощи полученного графа кратчайших путей между двумя произвольными точками пространства. Стадия обучения, в свою очередь, делится на два основных шага: генерация вершин графа и соединение этих вершин ребрами (см. Рис. 2)

Описанный подход очень быстро доказал свою эффективность для систем с большим количеством степеней свободы [12], и впоследствии были найдены применения данного метода и в других областях. Одной из этих областей является моделирование движения молекул. Одной из первых работ, использующих данный подход, было исследование пространственного расположения лигандов относительно белков [13]. Далее было множество различных реализаций для моделирования сворачивания протеинов [14][2][1]. Различия состояли в выборе энергетической функции, а также в выборе политики генерации исходного набора точек.

3.2. Энергетическая функция

Как упоминалось ранее, вершинами карты дорог являются точки в конфигурационном пространстве, которые представляют собой допустимые конфигурации. Допустимость конфигурации чаще всего (и в данной работе в частности) определяется значением потенциальной энергии. Обычно общая энергетическая функция выглядит следующим образом [10]:

$$E_{total} = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\Theta(\Theta - \Theta_{eq})^2 + \sum_{dihedrals} K_\phi(1 - \cos(n\phi)) + \sum_{atoms}^{i<j} \left[{A_{ij}}/{R_{ij}^{12}} - {B_{ij}}/{R_{ij}^{6}} + {q_i q_j}/{\epsilon R_{ij}} \right]$$

где первые три слагаемых описывают непосредственно суммарную энергию связей молекулы, а последнее слагаемое описывает взаимодействие каждой пары атомов и включает в себя энергию Ван-дер-Ваальса и электростатическое взаимодействие. Однако чаще всего вычисление значения приведенной выше функции оказывается крайне затратным, а большинство слагаемых несут в себе слишком маленький вклад, и поэтому во многих реализациях функция упрощается вплоть до следующего вида [2]:

$$E_{total} = \sum_{restraints} K_d \left[\sqrt{(d_i - d_0)^2 + d_c^2} - d_c \right] + \sum_{atoms}^{i<j} \left[{A_{ij}}/{R_{ij}^{12}} - {B_{ij}}/{R_{ij}^{6}} \right]$$

где первое слагаемое представляет собой ограничение, налагаемое на расстояние между соответствующими атомами водорода и кислорода (d_i), а второе слагаемое, как и ранее, энергию Ван-дер-Ваальса. Значения констант следующие: K_d = 100 КДж/моль, d_0 = d_c = 2Å, а значения A_{ij} и B_{ij} определяются из следующей таблицы:

Пара атомов	A	B
H..H	**290**	**1.07**
O..O	**145834**	**328**
N..N	**3952850**	**2556**
C..C	**1200965**	**425**
H..O	**2913**	**241**

Таб. 1

Причем коэффициенты для остальных пар атомов выражаются как среднее геометрическое соответствующих коэффициентов, например: $A_{O..N} = \sqrt{A_{O..O} \cdot A_{N..N}}$

3.3. Генерация вершин графа

После того как определена энергетическая функция, можно более конкретно описать стратегию выбора точек конфигурационного пространства в качестве вершин. Сначала случайным образом генерируются все координаты точки в соответствующем интервале (в данном конкретном случае все координаты – значения от 0 до 2π). Следующим шагом моделируется конфигурация протеина с соответствующими значениями торсионных углов. Затем вычисляется потенциальная энергия E полученной конфигурации. Данная конфигурация считается допустимой с вероятностью, вычисляемой следующим образом [13]:

$$P = \begin{cases} 0, \text{если } E > E_{max} \\ \dfrac{E_{max} - E}{E_{max} - E_{min}}, \text{если } E_{max} > E > E_{min} \\ 1, \text{если } E_{min} > E \end{cases}$$

где E_{min} и E_{max} - некоторые заранее определенные константы. Таким образом поощряются конфигурации с низкой потенциальной энергией, но разделение значений энергии на высокую и низкую происходит не скачком в одной точке, а плавно на отрезке от E_{min} до E_{max}.

Довольно часто, с целью получения более связного графа и более точного ответа на запрос, часть точек генерируется в непосредственной близости от точки, обозначающей свёрнутое состояние (этой информацией мы обладаем априори).

При реализации значения констант выбирались исходя из значения энергетической функции для исходной конфигурации.

3.4. Соединение вершин графа рёбрами

На данном этапе построения карты дорог предпринимаются попытки соединить каждую вершину графа с некоторым количеством ближайших (в некотором смысле, обычно здесь рассматривается Евклидово расстояние в конфигурационном пространстве) вершин. Обычно процесс продолжается либо до тех пор, пока не наберется достаточное количество соседей, либо

пока не будут просмотрены все возможные вершины (при этом слишком далёкие вершины можно даже не рассматривать, так как это очень затратно и слишком часто не приводит к успеху).

Попытка соединения вершин происходит следующим образом. Прямолинейный отрезок между этими вершинами делится на достаточное количество маленьких частей (здесь как параметр может выступать максимальная длина этого маленького участка), затем каждая промежуточная точка проверяется на доступность, и только если все промежуточные конфигурации оказались доступны, то и весь отрезок считается допустимым.

Ребра графа являются направленными и каждому допустимому отрезку соответствуют два ребра в графе, и в общем случае вес первого отличается от веса второго. Оба этих значения вычисляются одновременно с проверкой допустимости следующим образом. Пусть c_1, c_2 … c_n – начальная, все промежуточные и конечная конфигурации вдоль отрезка. Обозначим за ΔE разницу в энергии между двумя соседними конфигурациями, то есть $\Delta E_i = E(c_{i+1}) - E(c_i)$, $0 \leq i < n$,. Тогда вероятность перехода молекулы из конфигурации c_i в c_{i+1} можно вычислять следующим образом[2]:

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}}, \text{если } \Delta E_i > 0 \\ 1, \quad \text{иначе} \end{cases}$$

или, как альтернативный вариант, следующим образом [13]:

$$P_i = \frac{e^{\frac{-\Delta E_i}{kT}}}{e^{\frac{-\Delta E_i}{kT}} + e^{\frac{\Delta E_{i-1}}{kT}}}$$

С помощью полученных вероятностей перехода вес ребра вычисляется как сумма логарифмов этих вероятностей (со знаком «-», так как вероятности < 1), то есть

$$\omega(c_1, c_n) = \sum_{i=0}^{n-1} -\log P_i$$

Вычисление веса обоих рёбер может производиться одновременно.

3.5. Ответ на запрос

После построения карты дорог всё готово для обработки запросов. Запросы носят характер «переход из одной конфигурации в другую», что в конфигурационном пространстве представляется как запрос «путь из точки A в точку B».

Процесс вычисления ответа на такой запрос делится на несколько этапов. Сначала найдем несколько вершин графа, доступных из A (это можно сделать так же, как и в предыдущем пункте), обозначим это множество через C_A и запомним веса путей из точки A в каждую из этих вершин. Далее построим аналогичное множество C_B для точки B, только в этом случае запомним веса путей идущих из этого множества в точку B. Затем при помощи стандартного алгоритма Дейкстры (поиск кратчайших путей из одной вершины во все остальные в ориентированном взвешенном графе), применив его для каждой вершины из C_A, получим длины кратчайших путей от каждой из вершин множества C_A в каждую вершину множества C_B. Затем выберем вершины X из C_A и Y из C_B так, чтобы

$$d(A,X) + d_{PRM}(X,Y) + d(Y,B) = \min_{P \epsilon C_A, Q \epsilon C_B} (d(A,P) + d_{PRM}(P,Q) + d(Q,B))$$

где d_{PRM} означает длину пути в графе.

4. Особенности реализации

В качестве платформы для реализации был выбран язык Java и среда разработки IntelliJ IDEA.

4.1. Входные и выходные данные

3D-модели протеинов

В качестве входных и выходных данных были выбраны файлы формата .pdb (Protein Data Bank). На данный момент, этот формат поддерживается большинством программ для просмотра 3D-моделей молекул (QuteMol, JMol и др.). В данный формат переведено подавляющее большинство экспериментально полученных 3D-моделей протеинов. Подробное описание текущей версии формата можно скачать с веб-страницы [15].

Была разработана библиотека для работы с файлами данного формата: реализовано чтение исходных моделей и запись полученных результатов. Запись производится в несколько сокращенном варианте.

В разработанной модели читается и записывается следующая информация о протеине:

- техническая информация, такая как название, ключевые слова, авторы данной 3D-модели и т. п.
- информация о последовательности аминокислотных оснований
- геометрические координаты основных атомов каждого основания (N, H, CA, C, O, см. Рис. 1)
- геометрические координаты главного атома боковой цепи (CB)

При ответе на запрос в выходной файл записывается последовательность 3D-моделей, где первая модель – исходное состояние, а последняя модель – конечное состояние. Полученный файл можно просматривать, к примеру, при помощи программы JMol.

Карта дорог

Также в представленной реализации имеется возможность сохранения в файл построенной карты дорог. Это сделано из тех соображений, что построение достаточно подробных карт дорог для больших протеинов (это тысячи, а иногда даже десятки тысяч точек в конфигурационном пространстве) может занимать несколько часов, и не имеет смысла повторять данный процесс для нового набора запросов. Намного проще сохранить построенную модель в файл и по требованию загружать эту модель в оперативную память.

4.2. Карта дорог

Общее описание

Так как метод построения вероятностной карты дорог является самостоятельным и не связан с решением конкретной задачи, была разработана отдельная библиотека, в которой реализованы основные составляющие данного подхода:

- отдельный модуль для работы с ориентированными графами
- конфигурационное пространство и методы для работы с ним
- построение карты дорог в конфигурационном пространстве
- выполнение запросов к карте дорог

Все настраиваемые величины и методы вынесены в отдельный интерфейс.

Настраиваемые параметры:

- размерность пространства
- допустимые интервалы и цикличность для каждой координаты
- количество генерируемых вершин
- минимальная желаемая степень вершины графа

Настраиваемые методы:

- генерация набора случайных допустимых конфигураций
- соединение двух вершин рёбрами и определение весов этих рёбер

Для решения других задач с помощью вероятностной карты дорог достаточно реализовать предложенный интерфейс.

Ориентированный граф

Граф в данной реализации задаётся списком вершин и списком рёбер, но также одновременно со списком рёбер для каждой вершины дополнительно хранятся 2 списка – список исходящих рёбер и список входящих рёбер. Это позволяет за счёт использования дополнительной памяти упростить навигацию по графу, тем самым ускорить работу алгоритма Дейкстры в частности и процесс ответа на запрос в целом.

Так как полученный граф почти всегда будет сильно разреженным (количество вершин обычно несколько тысяч, степень каждой вершины редко больше 50), то в данном случае целесообразно реализовать алгоритм Дейкстры при помощи кучи для хранения текущих расстояний до еще непросмотренных вершин. Такая реализация даст сложность работы алгоритма порядка O(NlogN + ElogN), где N – количество вершин, а E – количество рёбер, тогда как простая реализация (с прямым поиском ближайшей непросмотренной вершины) имеет сложность порядка $O(N^2 + E)$.

Конфигурационное пространство

Конфигурационное пространство реализовано таким образом, что имеется возможность рассматривать некоторые координаты как циклические. Это является очень важным моментом при решении поставленной задачи, так как в данном случае все координаты представляют собой углы от 0 до 2π.

Например, если две точки сильно различаются лишь в одной координате и у одной из точек значение этой координаты близко к 0, а у второй – близко к 2π, то в случае классического пространства эти точки находятся довольно далеко друг от друга, когда как на самом деле они очень

близки (см. Рис. 3). В данной же реализации будет рассматриваться именно тот отрезок в пространстве, который в реальности является кратчайшим.

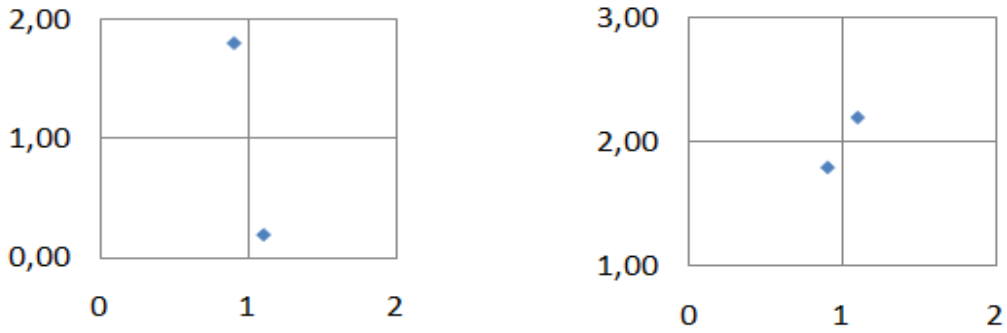

Рис. 3 Точки (1.1π, 0.2π) и (0.9π, 1.8π)

Если говорить подробнее о реализации данной возможности, то она достигается следующим не очень сложным образом. Для каждой координаты конфигурационного пространства хранится 2 числа, обозначающие верхнюю и нижнюю границы интервала, и одно булевское значение, имеющее значение true в том случае, если данная координата является циклической. Теперь, например, при расчете (действительно кратчайшего) расстояния между двумя точками по Евклидовой метрике достаточно при вычислении разности по каждой циклической координате сравнивать эту разность с половиной длины всего интервала. И если эта разность будет больше половины интервала, то можно одну из точек сдвинуть вдоль координаты на длину интервала, уменьшив при этом расстояние. Остальные детали, касающиеся циклических координат, реализованы подобным же образом.

4.3. Реализация

Генерация вершин графа

Оптимальная конфигурация известна априори, и так как часто запросы будут содержать данную конфигурацию в качестве конечной точки, то является логичным тот факт, что достаточно большую часть всех вершин графа нужно генерировать в непосредственной близости от данной точки. Разнообразные стратегии генерации были предложены и в работах [2][1]

В данной работе была реализована собственная политика генерации вершин графа. Далее здесь под расстоянием будет понимается следующая метрика (обычно обозначаемая как L_∞):

$$d(x,y) = \max_i |x_i - y_i|$$

На первом шаге $^1/_{12}$ всех точек генерируется равномерно на расстоянии от центральной точки (соответствующей оптимальной конфигурации) не более 0.1π. Далее на каждом k-м шаге следующая $^1/_{12}$ часть всех точек генерируется равномерно на расстоянии не более $0.1k\pi$ вплоть до k = 9 (см. Рис. 4). На последнем этапе оставшаяся $^1/_4$ всех точек генерируется равномерно по всему конфигурационному пространству (что в действительности означает равномерное распределение на расстоянии от центра не более π). На всех этапах точка может быть принята лишь в случае приемлемого значения энергетической функции (для точек с промежуточной энергией – с некоторой вероятностью, см. 3.3)

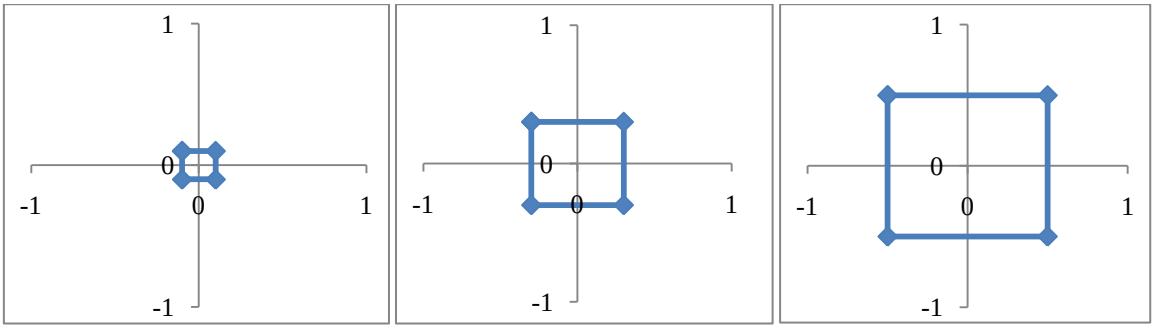

Рис. 4 Области равномерного распределения для k=1,3 и 5

Соединение точек

Соединение вершин графа рёбрами происходит ровно тем способом, который описан в 3.4. Для каждой следующей вершины составляется список ближайших к ней вершин в смысле обычной Евклидовой метрики (или также известной как L_2):

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Затем при помощи дробления отрезка на промежуточные состояния и вычисления энергетической функции определяется, можно ли данные вершины соединить ребром, и если да - то какой вес будет в каждом из двух направлений.

Стоит дополнительно отметить, что порог энергетической функции на стадии соединения значительно выше, чем на стадии генерации вершин.

Энергетическая функция.

Основной расчёт энергетической функции производился по формуле, уже упоминавшейся выше:

$$E_{total} = \sum_{restraints} K_d \left[\sqrt{(d_i - d_0)^2 + d_c^2} - d_c\right] + \sum_{atoms}^{i<j} \left[{A_{ij}}/{R_{ij}^{12}} - {B_{ij}}/{R_{ij}^{6}} \right]$$

Однако перед непосредственным вычислением значения выполняется несколько проверок. Если какие-либо два атома CB (обозначающие в грубом приближении боковую цепь) находятся на расстоянии менее чем 1Å, то вычисление не производится, и функция возвращает очень большое число (превосходящее порог на этапе соединения). Если же все расстояния превосходят 1Å, но есть расстояние, меньшее 2.4Å, то данная конфигурация принимается на этапе соединения, но не принимается на этапе генерации, то есть просто возвращается значение, превосходящее порог на этапе генерации, но меньшее значения порога на этапе соединения. Это не внесёт значительной неточности, так как ребро с такой промежуточной конфигурацией будет иметь очень большой вес и может быть полезным лишь для уменьшения количества компонент связности.

Пространственная геометрия

На этапах генерации вершин графа и соединении вершин рёбрами требуется постоянно вычислять энергетическую функцию в заданной точке конфигурационного пространства. Эту операцию можно разделить на две составляющие – построение молекулы по заданным координатам и

непосредственно вычисление потенциальной энергии полученной молекулы. Если вторая часть подробно описана в предыдущем пункте, то первая представляет собой непростую геометрическую задачу.

В целях решения данной задачи предложена эффективная реализация методов для работы с точками в трёхмерном пространстве. Построение происходит следующим образом. Первые три точки полипептидной цепи считаются фиксированными, также известны длины всех связей, плоские углы, а также пространственные углы, позволяющие определять положение боковых атомов (таких как водород и кислород) относительно основной цепи. Трёхмерные координаты атомов основной цепи вычисляются постепенно, каждый следующий атом однозначно восстанавливается по трём предыдущим атомам цепи, длине связи, плоскому углу и торсионному углу (который либо фиксирован, либо является значением следующей координаты в конфигурационном пространстве). Затем по полученной основной цепи и известным фиксированным параметрам однозначно восстанавливаются координаты боковых атомов.

5. Полученные результаты

Для проведения экспериментов была выбрана молекула белка GB1 (иммуноглобулин, см. [15]). Молекула состоит из 56 оснований, а значит размерность конфигурационного пространства – 110.

Для исследования было построено 2 карты дорог различной степени детализации. Одна содержала 2000 вершин, вторая – 10000. В первом случае генерация карты дорог заняла примерно 50 минут, во втором случае – около 6 часов. Построение производилось несколько раз. Лучшие полученные результаты кратко представлены в Таб. 2

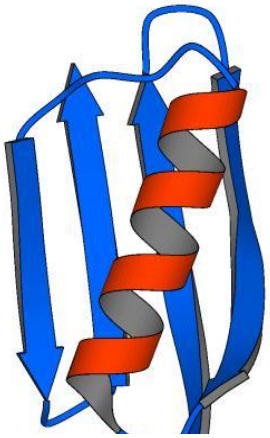

Рис. 5

#V	#E	#MaxCC	100Q	%ans	Avg. path
2000	55000	1994	400	98	6,5
10000	470000	9992	2500	99	8,5

Таб. 2

Здесь #V – количество вершин, #E – количество рёбер, #MaxCC – средний размер максимальной компоненты связности графа, 100Q – время в секундах на обработку 100 произвольных запросов, %ans – доля найденных путей, Avg. Path – средняя длина пути в графе, являющегося ответом на запрос. Разброс значений составил не более 10% для случая 2000 вершин и не более 5% для 10000. Для сравнения приведем подобные результаты из работы [2]:

#V	#E	#MaxCC	100Q	Avg. path
2000	50000	1998	5000	9
10000	260000	9997	39000	6

Таб. 3

Стоит отдельно отметить довольно сильное различие по времени выполнения запроса. Эта разница скорее всего объясняется тем, что в данной работе использовалась классическая реализация алгоритма Дейкстры, время работы которой порядка $O(N^2 + E)$.

Далее сравнивались результаты ответа на запрос к обеим построенным картам дорог. В качестве примера приведем статистику при выполнении одного и того же запроса. На Рис. 6 и Рис. 7 показано изменение энергии в промежуточных точках полученного пути (точка 0 обозначает начальное состояние, точки, соответствующие нижним делениям – вершины в графе, последняя точка – искомое состояние, промежуточные энергии вычислялись в 20 точках каждого прямолинейного отрезка)

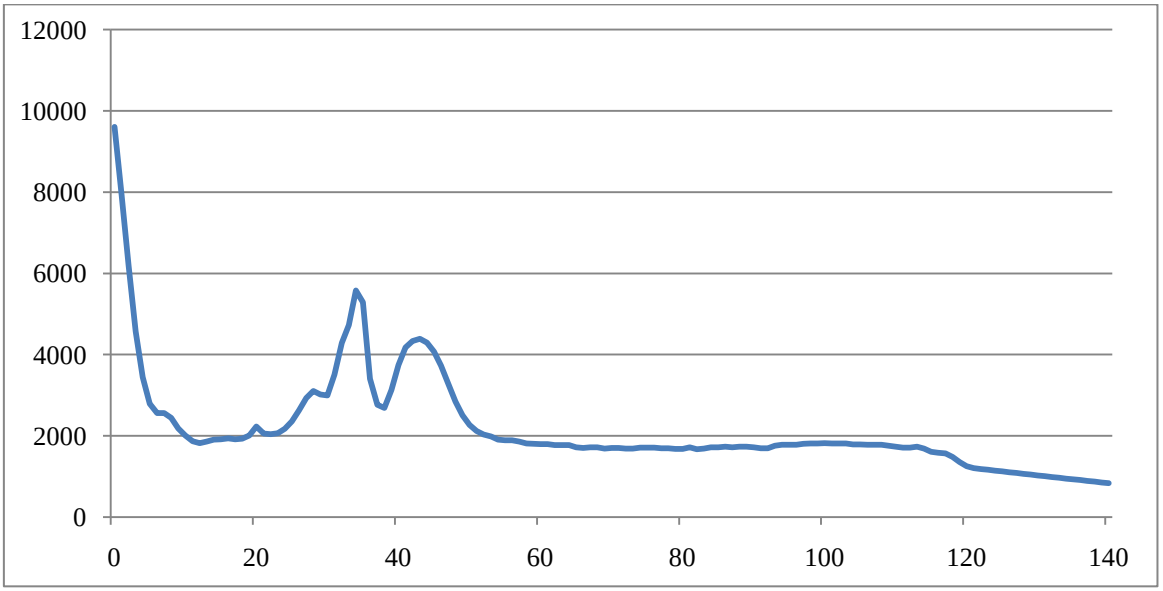

Рис. 6

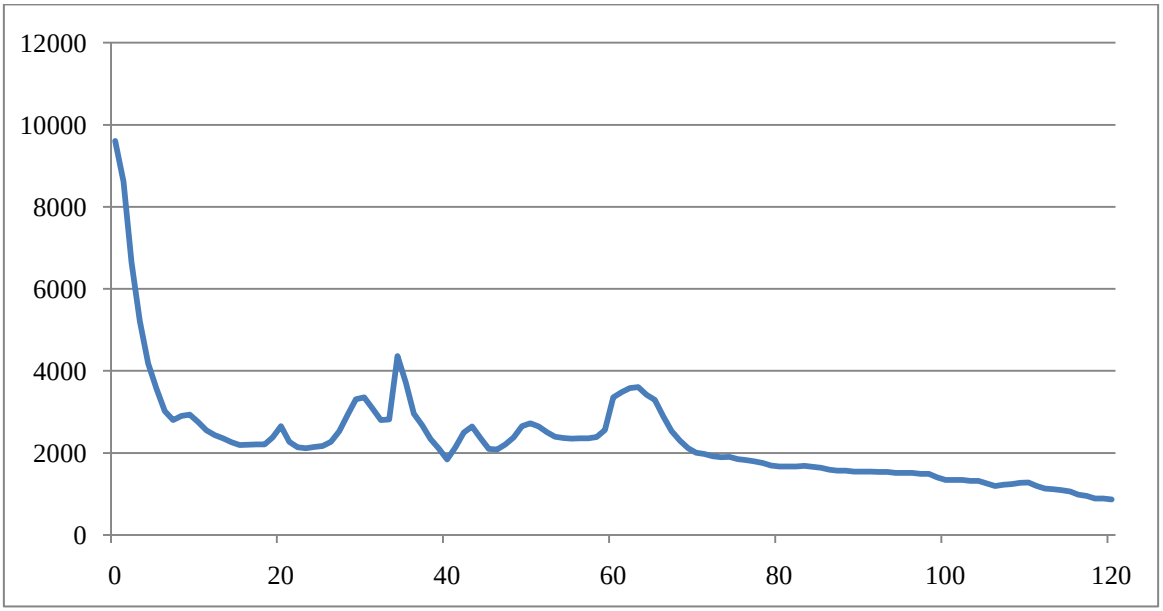

Рис. 7

Также в качестве дополнительной проверки корректности работы программы было исследовано, в какой последовательности в полученных ответах формируются элементы вторичной структуры молекулы – α-спираль

и 4 β-листа (обозначаемые β1, β2, β3 и β4) (см. Рис. 5). Вручную было проверено 20 ответов на произвольные запросы. Во всех 20ти случаях первой формируется α-спираль. Также в большинстве случаев (в 15 случаях из 20) следующим этапом формируется пара листов β3-β4, и только затем пара β1-β2. А в оставшихся 5 случаях пара β1-β2 формируется либо на несколько шагов раньше, либо одновременно с β3-β4. Полученный результат подтверждается другими исследованиями на эту тему [16].

Дополнительно продемонстрируем работу нашего подхода к генерации начального множества вершин. На Рис. 8 изображена проекция 5% всех точек (для случая карты дорог на 10000 вершин) на первые две координаты конфигурационного пространства (от 0 до 2π). Как видно из рисунка, довольно большое множество точек сосредоточено в непосредственной близости от координат естественной конфигурации (1,331π и 0,716π). Однако стоит отметить, что достаточное множество точек равномерно распределено по всему пространству. Похожую картину можно наблюдать и в [2].

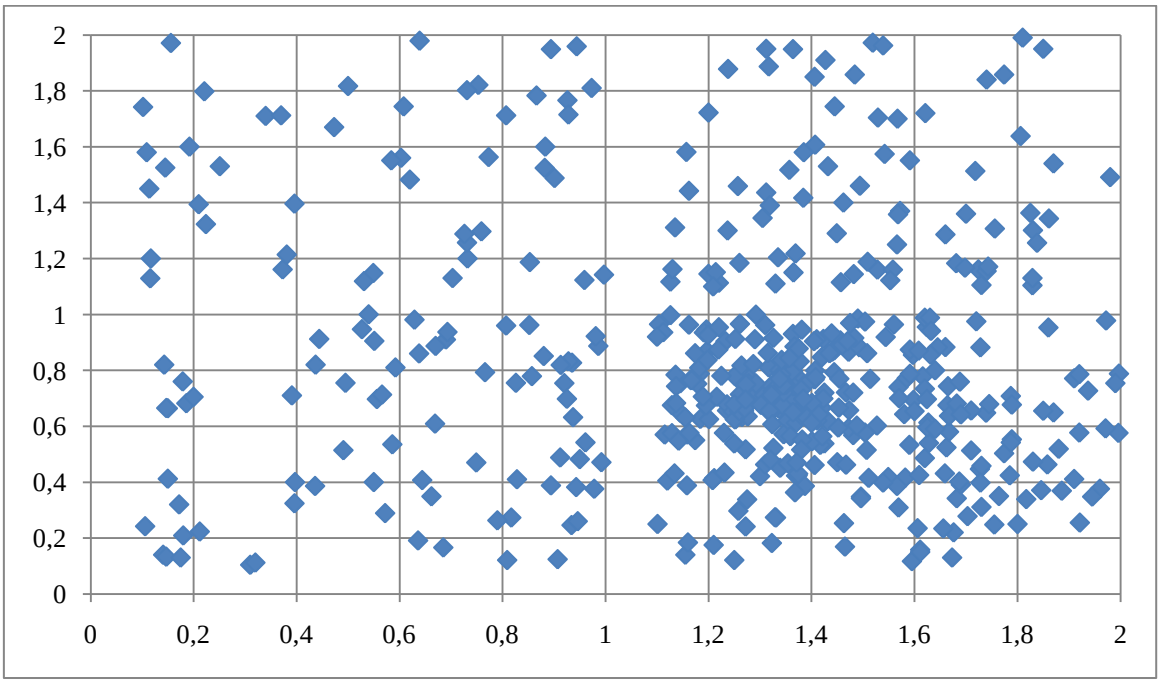

Рис. 8

6. Заключение

Результатом данной работы является программная реализация метода вероятностной карты дорог для анализа поведения белковых молекул. Полученная реализация оформлена в виде консольного приложения. Проведены эксперименты, по результатам которых показана применимость полученной реализации для решения поставленных задач.

Разработана библиотека для построения карты дорог. Полученная реализация является независимой и может быть использована для решения других подобных задач. Реализованы методы для чтения и записи файлов формата .pdb, а также библиотека для работы с пространственными конфигурациями протеинов.

Дальнейшая работа по данной тематике состоит в выборе наиболее адекватных параметров, таких как, например, энергетическая функция, стратегии генерации и соединения вершин. Также в дальнейшем возможен переход от вероятностной карты дорог к стохастической карте дорог (см. 2.4) и решение связанных с данным подходом задач.

Список литературы

1. S. Thomas, G. Song, N. M. Amato. Protein folding by motion planning // Physical biology. 2005, 2, S148–S155 pp.

2. N. M. Amato, G. Song. Using motion planning to study protein folding pathways // J. Comput. Biol. 2002, 9, 149-168 pp.

3. J. M. Haile. Molecular Dynamics Simulation: Elementary Methods. New York : John Wiley & Sons, 1992.

4. A. R. Leach. Molecular Modelling: Principles and Applications. Essex, England : Longman, 1996.

5. Folding@home distributed computing // Folding@home. 15.05.2011 http://folding.stanford.edu

6. J. Shimada, E.L. Kussell, E.I. Shakhnovich. The folding thermodynamics and kinetics of crambin using an all-atom monte carlo simulation // J. Mol. Biol. 2001, 1, 79–95 pp.

7. R. Czerminski, R. Elber. Self avoiding walk between two fixed points as a tool to calculate reaction paths in large molecular systems // Int. J. Quant. Chem. 1990, 24, 167-186 pp.

8. G. Henkelman, G. Jhannesson, H. Jnsson. Methods for finding saddle points and minimum energy paths // Progress on Theoretical Chemistry. Kluwer Academic Publishers, 2000.

9. A. Ghosh, R. Elber, H. A. Scheraga. An atomically detailed study of the folding pathways of protein a with the stochastic difference equation // PNAS. 2002, 16, 10394–10398 pp.

10. M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, C. Varma. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. // J Comput Biol., 2003, 10, 257-281 pp.

11. L. E. Kavraki, P. Svestka, J.-C. Latombe, M. H. Overmars Probabilistic roadmaps for path planning in high-dimensional configuration spaces // IEEE Transactions on Robotics and Automation. 1996, 12, 566–580 pp.

12. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun. Principles of Robot Motion. Theory, Algorithms, and Implementations. The MIT Press, 2005.
13. A. P. Singh, J.-C. Latombe, D. L. Brutlag. A motion planning approach to flexible ligand binding // 7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB). 1999, 252-261 pp..
14. G. Song, N. M. Amato. A motion planning approach to folding: from paper craft to protein folding // Proc. IEEE Int. Conf. Robot. Autom. (ICRA). 2001, 948-953 pp.
15. Worldwide Protein Data Bank. 15.05.2011. http://www.wwpdb.org/docs.html
16. E. L. McCallister, E. Alm, D. Baker. Critical role of β-hairpin formation in protein G folding // Nat. Struct. Biol. 2000, 7, 669–673 pp.