Санкт-Петербургский Государственный Университет
Математико-Механический факультет

кафедра системного программирования

Сервер морфинга протеинов

Курсовая работа студента 445 группы
Лушникова Андрей Сергеевича

Научный руководитель: **К.В.Вяткина**

Санкт-Петербург
2010

Оглавление

Введение

В данной работе представленна реализация веб-сервера для морфинга протеинов. Несмотря на то, что на данный момент есть подобные решения, необходимость в создании аналога обусловлена наличием более современных подходов к процессу как выравнивания, так и непосредственно морфинга протеинов. Наличие подобного сервера предоставило бы отличную платформу для новых алгоритмов и подходов для морфинга протеинов.

На данный момент наиболее популярным сервером для морфинга протеинов является сервер Gershtein'a[1], запущенный в 2000 году. Целью работы была реализация сервера, не уступающего по функциональности существующему аналогу, и реализующему алгоритм, предложенный Павлом Певзнером, Natalie Castellana и их коллегами из University California, San-Diego.

Данный проект разрабатывается в сотрудничестве с Павлом Певзнером, лабораторией биоинформатики Санкт-Петербурга и UCSD.

1 http://molmovdb.org/cgi-bin/submit.cgi

Терминология

Протеины, они же белки – органические вещества, состоящие из остатков аминокислот, соединённых пептидной связью. В последнее время протеины широко изучаются по всему миру, что связано с чрезвычайно важными и разнообразными функциями белка в организме. Один из важных параметров белка – пространственное строение его пептидной цепи, т.е. набор пространственных координат составляющих белок атомов.

В последнее время огромное количество белков было проанализированно с помощью рентгеновского анализа. Результаты распространяются в формате pdb – формат содержит местоположение атомов белка в трёхмерном пространстве.

Процесс спонтанного сворачивания белка в уникальную третичную структуру называется фолдингом белка и на данный момент плохо изучен. Однако, зная начальную и конечную структуру белка, можно попробовать смоделировать процесс фолдинга. Моделирование изменения третичной структуры белка называется морфингом.

Морфинг, как и любое моделирование, оперирует не самими белками, которые являются исключительно сложными веществами, а их абстракциями. Третичная структура в биоинформатике принятно моделировать с помощью ломанной линии в трёхмерном пространстве. Именно этот способ использовался в данной работе.

Морфинг двух протеинов зависит от их взаимного расположения в пространстве. Поэтому два протеина изначально “выравнивают” друг относитльно друга. В данной курсовой работе это сделано популярным способом минимизации среднеквадратичного отклонения расстояния между аминокислотными остатками протеинов.

Обзор технологий

Сервер

При разработке проекта было проведено предварительное исследование серверных технологий. Рассматривались следующие варианты:

- java ee
- python + django
- ruby + ruby on rails

Java EE широко используется в связке с Inversion-of-control фреймворком Spring. Системы, построенные на этой технологии, отлично масштабируются, и обладают превосходной производительностью. Этим подходом успешно пользуются таки крупные интернет-сервисы, как yandex. Однако основным их недостатком можно считать большую трудоёмкость разработки, что, в условиях достаточно небольшого ожидаемого числа клиентов, не оправдывает себя.

Языки Python и Ruby в связке с соответствующими веб-фреймворками django и ruby on rails представляются достаточно равноценными. Однако, ввиду таких факторов, как наличие высоконадёжного облачного сервиса для хостинга ruby-приложений heroku.com, а так же ввиду субъективных ощущений, выбор пал на язык ruby и веб-фреймворк ruby on rails.

Задача чтения и разбора pdb-файлов, загружаемых пользователем, была решена с помощью rubygem'a “bio” и его модуля “Pdb”. Хранение всех данных пользователя организовно в реляционной базе данных[2]. Это не является оптимальным решением, т.к. чтение pdb-файлов с диска происходит существенно быстрее, чем загрузка этой же информации из базы данных, однако heroku не позволяет загружать файлы в свой репозиторий из-вне.

Клиент

Клиентская часть ответственна за отображение морфинга двух состояний протеинов в трёхмерном пространстве. До недавних пор задача отображения трёхмерного web-контента повсеместно решалась с помощью таких технологий, как flash, java-апплеты, sylverlight, или даже с помощью написания отдельных платформо-зависивых тонких клиентов, которые необходимо устанавливать на компьютер пользователя. Все эти технологии обладают одним

2 На данный момент используется SQLite, однако Rails позволяет использовать любую популярную БД, поэтому в любой момент движок БД может быть изменён на более подходящий.

существенным недостатком: они требуют от пользователя дополнительных плагинов и/или технологий, которых у него вполне может и не быть. Так, например, сервер Gershtein'a использует апплеты для решения этих целей, а потому у большинства пользователей Linux возникают традиционные проблемы, выражающиеся в надписи “Missing Plugin”.

Однако 2 марта 2011 года вышла первая версия стандарта технологии WebGL, которая позволяет аппаратно просчитывать трёхмерную сцену и отображать её на html5 элементе canvas. Для этого не требуется никаких плагинов, технологию поддерживают основные браузеры: Opera, FireFox, Chrome, Safari.

Технология является исключительно низкоуровневой. Поэтому было решено воспользоваться сторонней библиотекой, предоставляющей возможность рендеренга 3D в браузере с использованием технологии WebGL. Для этих целей есть несколько решений: GLGE, SceneJS, SpiderJS,PhiloGL, однако выбор пал на библиотеку Three.js. Она обладает очень понятным открытым исходным кодом, выложенным на github, хорошей коллекцией примеров и irc-каналом разработчиков, готовых ответить на любые вопросы. Библиотека очень молодая и находится на стадии активной разработки, поэтому документация к ней, к сожалению, отсутствует.

Одним из бонусов, полученных при использовании библиотеки three.js, стала поддержка рендеринга 3D-сцены без webgl на software уровне. Таким образом была решена проблема, возникающая на комьютерах под управлением OS Linux с видеокартой от intel: на сегодняшний день на машинах такой конфигурации webgl, к сожалению, инициализировать не удаётся.

Алгоритмы

Морфинг

Для морфинга двух протеинов был реализован алгоритм из черновика статьи Павнла Певзнера и Natalie Castellana. Основная идея алгоритма заключается в построении некоторого графа переходов и использовании на нём динамического программирования[3].

Первая реализация была написана непосредственно на ruby, однако морфинг небольших протеинов длился на протяжении 25 минут, в то время как теоритические оценки указывали на время в несколько секунд. Было проведено исследование скорости работы ruby, в результате которого было выяснено, что ruby – исключительно медленный и едва ли подходит для реализации алгоритмов. Так, например, 4*10^6 операций цикла for с пустым телом выполняются

3 Детали алгоритма, к сожалению, описанию не подлежат, т.к. сам алгоритм ещё не опубликован.

3 секунды. Поэтому алгоритм был целиком переписан на ANSI C, а для связи с ruby-on-rails была написана ruby-обёртка над этим алгоритмом. В результате удалось достингуть времени выполнения алгоритма в 3 секунды для небольших протеинов в 80 аминокислотных остатков, и 7 секунд для достаточно больших протеинов размера 250 аминокислотных оснований.

Кроме того, реализован алгоритм, генерирующий промежуточные положения морфинга с помощью выпуклой комбинации. Этот алгоритм используется по умолчанию в текущей версии сервера.

Выравнивание

Задача выравнивания начального положения протеинов обычно трактуется как задача о минимизации среднеквадратичного отклонения расстония между аминокислотными остатками. Для решения этой задачи есть широкоизвестный алгоритм Кабша, опубликованного в 1976 году. Предложенный им алгоритм решает матричное уравнение с учётом всех возникающих крайних случаев. Однако в процессе изучения возможных подходов выбор пал на алгоритм QCP.

QCP – наиболее быстрый из известных алгоритм для решения задачи минимизации среднеквадратичного отклонения между двумя множества точек. Алгоритм определяет матрицу поворота 3x3, на которую надо повернуть второе множество точек, чтобы минимизировать среднеквадратичное отклонение. Основная оптимизация достигается за счёт использования свойств кватернеонов и вместо операций обращения матриц. Его реализация под лицензией BSD на языке ANSI C используется на данный момент.

Результаты

Результатом курсовой работы стал REST-сервис по морфингу протеинов, позволяющий пользователям загружать свои протеины в формате pdb на сервер, выбирать нужное число шагов морфинга, и просматривать результаты в трёхмерном пространстве без необходимости устанавливать какое-либо программное обеспечение. В режиме просмотра результатов пользователи могут переходить от текущего шага морфинга к следующему или предыдущему, включать непрерывную анимацию процесса, вращать, приближать и отдалять 3D-сцену. Для поддержки слабых компьютеров предусмотрены настройки детализации 3D-сцены. Все когда-либо проводимые запросы на морфинг протеинов доступны в архиве сайта и доступны каждому.

Альфа-версия сервиса доступна по адресу: http://simple-waterfall-188.heroku.com/

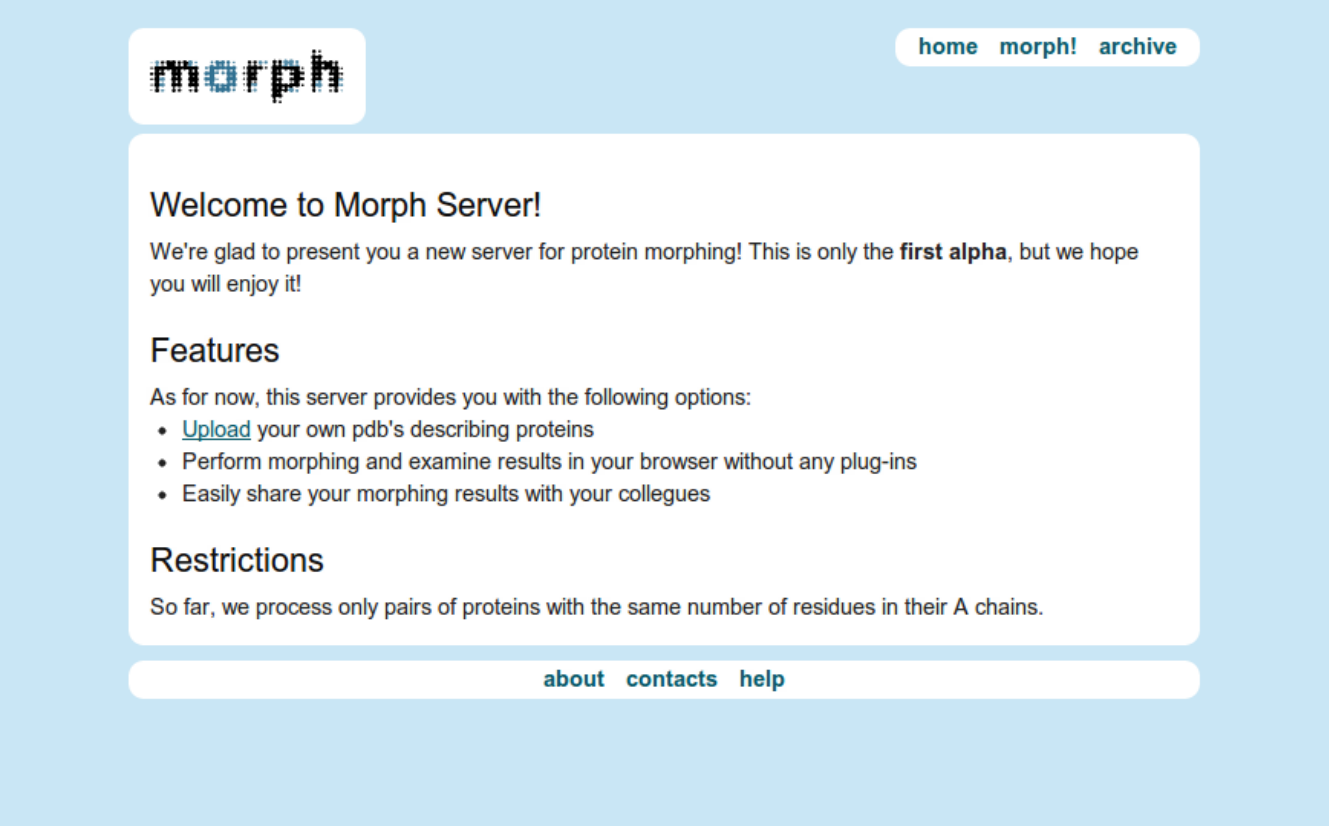

Рис. 1. Главная страница сервиса

Рис. 2. Создание нового морфинга

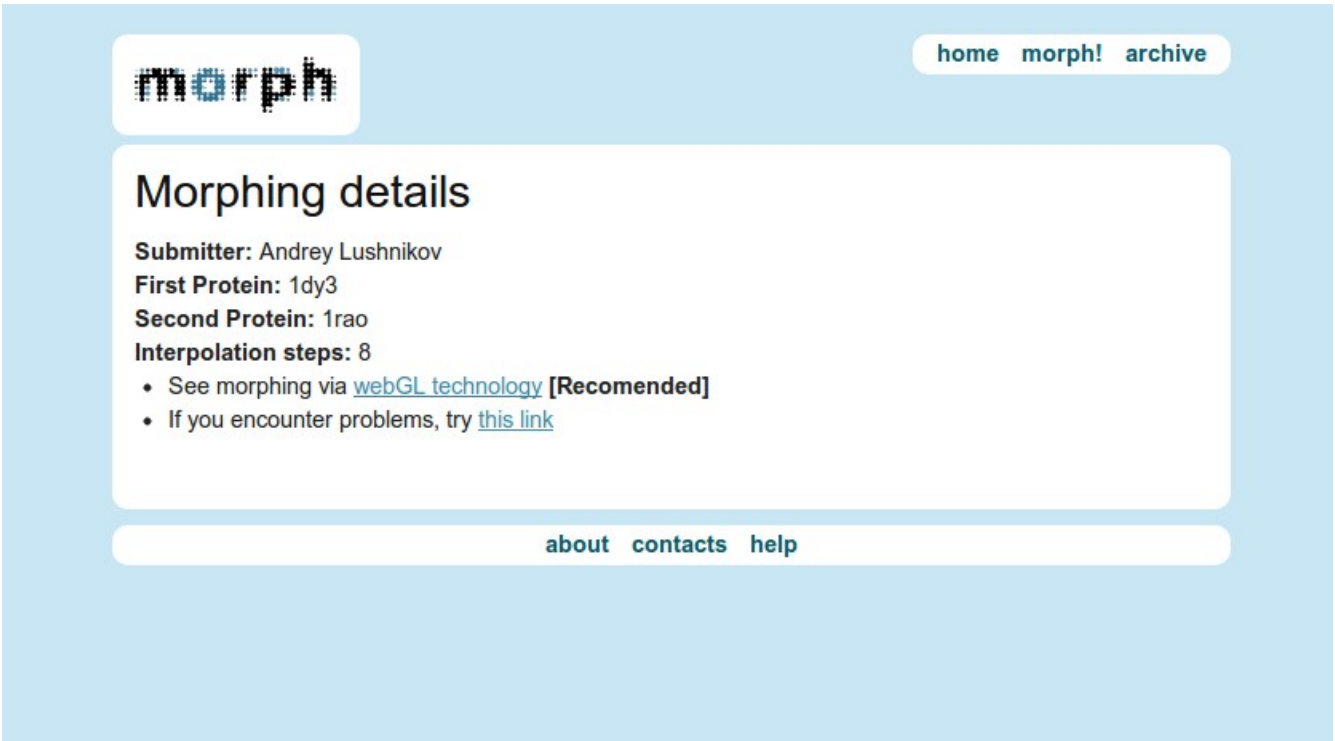

Рис.3. Информация о морфинге

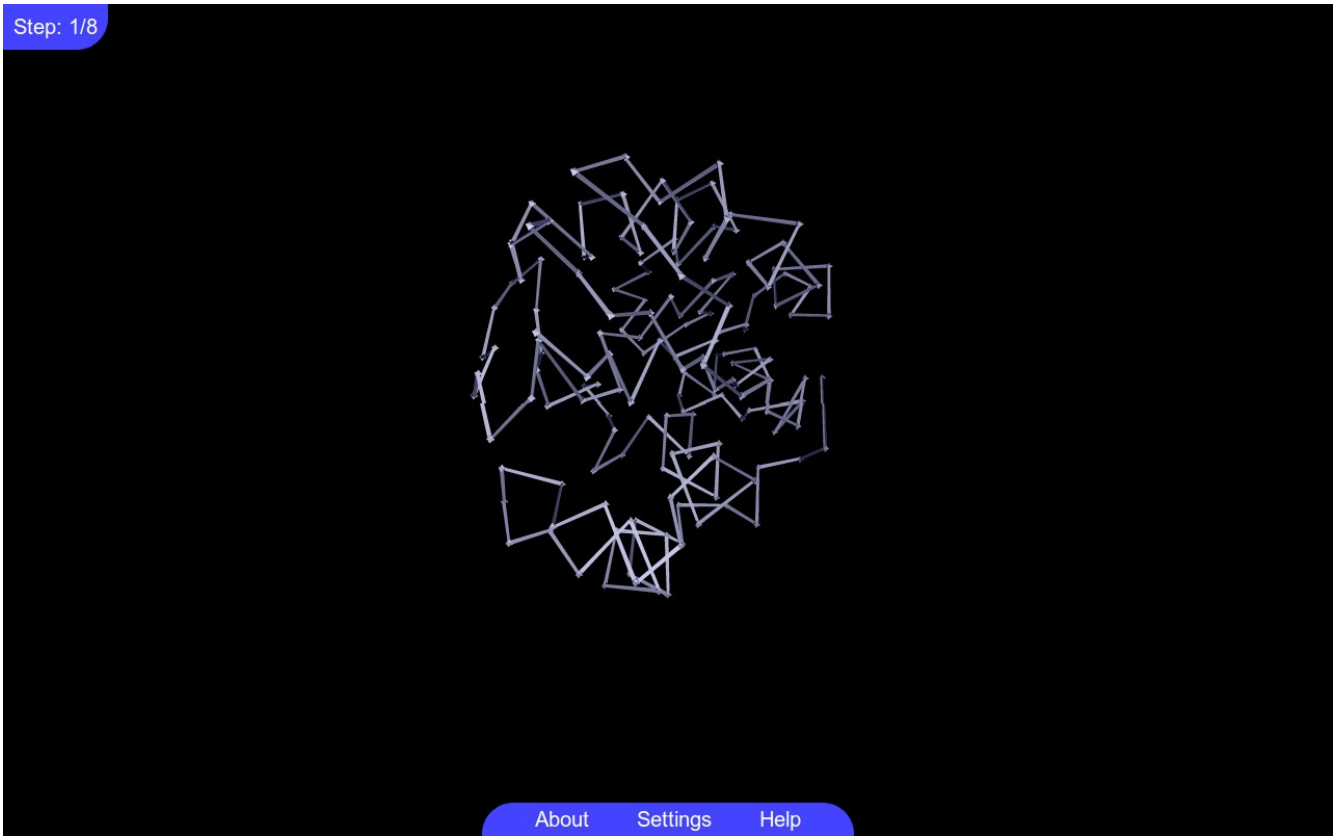

Рис.4. Просмотр морфинга в браузере

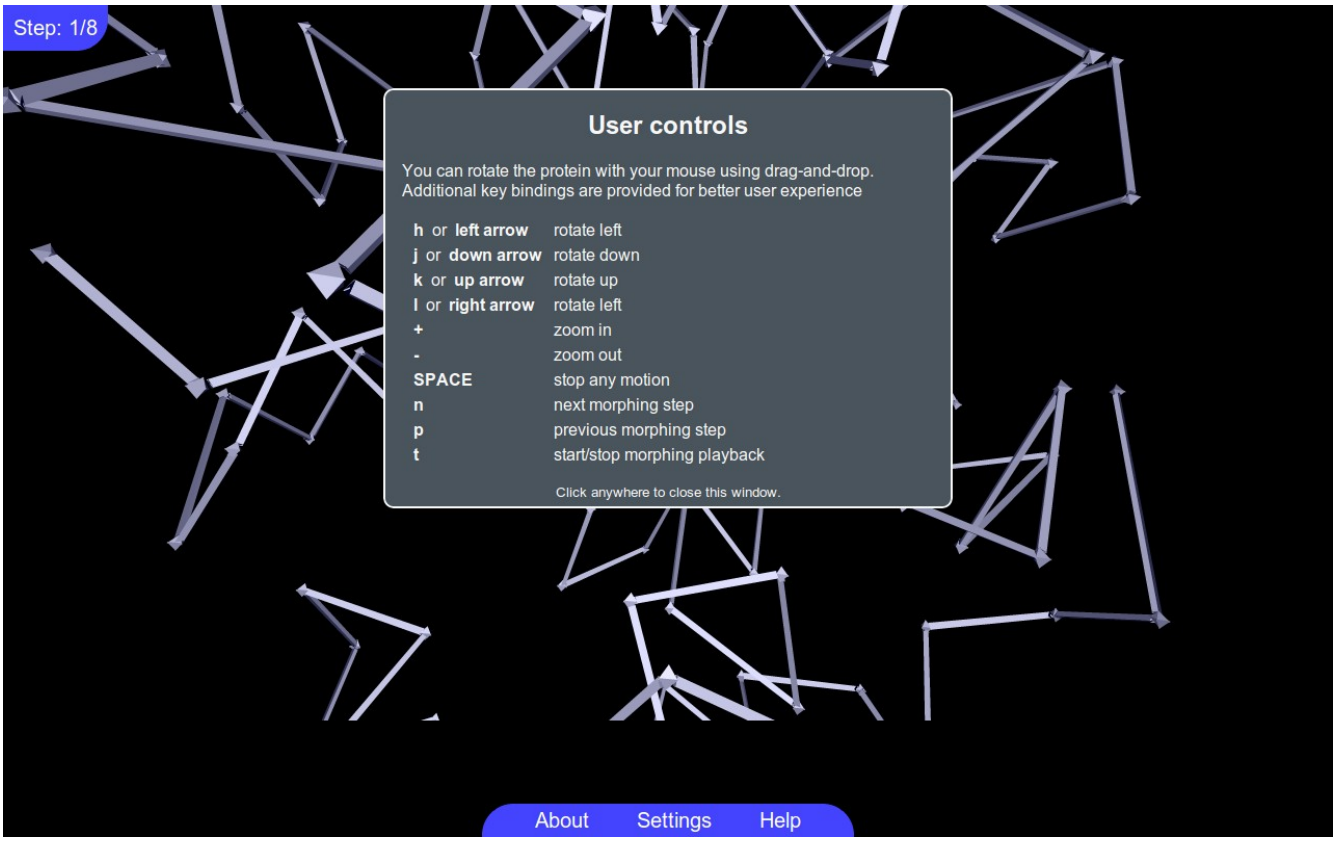

Рис.5. Окно помощи для просмотра

Пути развития

Основной упор в развитии сервера будет сделан на улучшении алгоритмов морфинга. Возможными направлениями для развития сервера будут:

1. Использование расстояния Фреше для выравнивания протеинов
2. Использование более совершенных алгоритмов морфинга протеинов
3. Хранение загруженных пользователями pdb-файлов на серверах Amazon с помощью сервиса Amazon S3

Кроме того, на развитие сервера повлияют многочисленные отзывы специалистов, использующих сервис.

Немаловажным является избавление сервиса от некоторый ограничений, присутствующих в текущей версии:

1. Невозможность морфинга протеинов с разным числом аминокислотных остатков
2. Невозможность выбора цепей протеинов

Так же немаловажным будет экспорт результатов морфинга в формат gif или ему подобный.

Список литературы

1. http://theobald.brandeis.edu/qcp/
2. https://github.com/mrdoob/three.js/
3. http://www.khronos.org/webgl/
4. Dave Thomas, Andy Hunt, “Programming in ruby”, 2000
5. Scott Chacon, “Pro Git”, 2009
6. http://ruby.railstutorial.org/ruby-on-rails-tutorial-book
7. John McCreesh, “Four days on rails”
8. Obie Fernandez, “The rails way”
9. David Flanagan, “The ruby programming language”
10. http://learningwebgl.com/blog
11. Aaftab Munshi, Dan Ginsburg, Dave Shreiner, “Open GL ES 2.0 Programming Guide”
12. http://en.wikipedia.org/wiki/Kabsch_algorithm
13. http://bioruby.open-bio.org/