МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ФАКУЛЬТЕТ

ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ Кафедра

биомедицинской информатики

Разработка среды для построения структур белковых комплексов

Курсовая работа

Рака Алексей Степановича
студента 3 курса,
специальность «Информатика»

Научный руководитель:
Хадарович А.Ю.

Минск, 2018

Оглавление

Введение

Выравнивание структуры белка – это ценный инструмент для фолдинга белков и классификации функций. Успех структурной геномики, направленной на экспериментальное определение трёхмерных структур тысяч репрезентативных белков, в решающей степени зависит от нашей способности разрабатывать точные инструменты для сравнения белковых структур. Однако, несмотря на свою исключительную важность, проблема по-прежнему не имеет быстрого и точного решения. В то время, как некоторые функции скоринга структурного подобия могут быть аппроксимированы за полиномиальное время, нет никакой процедуры для оптимизации любой меры структурного выравнивания. В своей обзорной статье о прогрессе в области сравнения структур Тейлор и коллеги пишут: “в сравнении структур у нас даже нет алгоритма, который гарантирует оптимальный ответ для пары структур”.

Существует несколько различных, но связанных между собой определений оптимального выравнивания двух белков. Некоторые методы определяют оптимальную суперпозицию, минимизирующую расстояние между выровненными атомами. Другие методы пытаются минимизировать разницу между внутриатомными расстояниями.

Недавно было введено несколько методов улучшения соответствия белковых структур, включая методы, основанные на фенотипической пластичности и метод гибких выравниваний последовательностью локальных преобразований.

Пожалуй, наиболее интуитивной и широко используемой мерой подобия двух белков является наибольшее количество атомов в двух структурах, которые могут быть совмещены друг с другом на заданное расстояние. Далее будем обозначать эту матрицу $CA \leq \sigma$, где $\sigma > 0$ обозначает порог расстояния в ангстремах.

Многие меры структурного выравнивая построены на $CA \leq \sigma$, включая GDT, AL0, MaxSub, CA-atoms < 3Å, Q-score и TM-score. Global Distance Test (GDT) обычно используется для оценки качества моделей в эксперименте CASP. Точность прогнозируемой модели в CASP измеряется оценкой GDT_TS, которая представляет собой среднее значение оценок GDT, вычисленных на нескольких порогах расстояния. Точнее,

$$\text{GDT\_TS} = \frac{(\text{GDT\_P1} + \text{GDT\_P2} + \text{GDT\_P4} + \text{GDT\_P8})}{4}$$

Где GDT_Pn определяет часть атомов CA, которые могут быть совмещены на расстоянии не превосходящем n ангстремов.

Одной из главных мер качества модели в LiveBench является CA-atoms < 3 Å(в наших обозначениях CA < 3). Из-за сложности оптимизации самой функции подсчёта очков LiveBench приближается к CA < 3 с помощью 3deval, программы, которая пытается максимизировать другую матрицу, а именно 3D-оценку.

CAFASP бенчмарк использует MsxSub для оценки качества прогнозов. MaxSub определяется как взвешенная доля остатков в модели, попадающих в пределы 3.5 Åот выровненных остатков в экспериментальной структуре.

Независимо от используемой системы подсчёта очков, основная трудность, которую должен преодолеть любой метод структурного выравнивания, – это бесконечное (несчётное) пространство всех возможных структурных суперпозиций. Чтобы обойти эту проблему, текущие исследования в этой области фокусируются на уменьшении размера пространства поиска путём перечисления относительно небольшого, но репрезентативного набора суперпозиций. Однако, хотя решения, полученные эвристическими подходами, частично точны, они никогда не гарантируются даже близкими к оптимальным решениями.

Очевидным способом устранения ограничений эвристических методов является разработка быстрого и точного метода максимизации $CA \leq \sigma$. Такой метод позволил бы точно вычислить ряд мер структурного выравнивания, включая GDT_TS, AL0 и MaxSum.

В данной работе рассмотрим алгоритм, который способен найти суперпозицию, которая достаточно близка к оптимальной суперпозиции. Более кон-

кретно, для любого заданного расстояния среза $\sigma > 0$ и любого $\varepsilon > 0$ алгоритм возвращает суперпозицию, которая помещает по крайней мере столько пар остатков на расстоянии не превосходящем $\sigma + \varepsilon$ сколько оптимальная суперпозиция содержит пар на расстоянии не превосходящем σ. В дополнение к относительно низкой временной сложности и способности к параллельным реализациям, данный алгоритм обеспечивает "качество решения которое определяется как разница между оценкой возвращённой суперпозиции и оценкой оптимальной суперпозиции. Метрика качества решения может использоваться для определения того, является ли возвращённая суперпозиция оптимальной суперпозицией (или необходим дургой более подробный поиск).

Данный алгоритм приближённого решения основан на схеме Колодного и Линиала для аппроксимации для класса непрерывных мер структурного подобия, работающей за полиномиальное время. Однако представленные здесь результаты не могут быть получены из их исследования, поскольку целевая функция $CA \leq \sigma$ не относится к категории оценочных функций описываемых Колодным и Линиалом функций. На самом деле, класс оценочных функций поддающихся методам Колодного и Линиала (а именно в классу функций удовлетворяющих условию Липшица) не включает в себя GDT_TS, AL0, MaxSum, TM-score, Q-score и некоторые другие меры структурного сходства белков.

Наконец, в данной работе решается проблема нахождения процедуры, которая возвращает оптимальную суперпозицию двух структур. Здесь представляется процедура, которая гарантированно вернёт оптимальное решение с вероятностью 1, т.е. для всех, кроме конечного значений, расстояний. Однако подчеркнём, что алгорим оптимального решения не работает за полиномиальное время. Так же известно, что это проблема является NP-трудной.

Глава 1

Пространственное выравнивание белков

Пространственное выравнивание — способ установления гомологии между двумя или более полимерными структурами на основании их трёхмерной структуры. Этот процесс обычно применяется к третичной структуре белков, но может также использоваться и для больших молекул РНК. В противоположность простому наложению структур, когда известно по крайней мере несколько эквивалентных аминокислотных остатков, пространственное выравнивание не требует никаких предварительных данных, кроме координат атомов.

Пространственное выравнивание подходит для сравнения белков с непохожими последовательностями, когда эволюционные отношения не могут быть установлены стандартными методами выравнивания последовательностей, но в этом случае необходимо принимать во внимание влияние конвергентной эволюции.

Пространственное выравнивание позволяет сравнивать две и более молекулы, для которых известны трехмерные структуры. Два основных метода их получения — рентгеноструктурный анализ и ЯМР-спектроскопия. Для пространственного выравнивания можно также использовать структуры, полученные методами предсказания структуры белка. Пространственные выравнивания особенно важны для анализа данных, полученных методами структурной геномики и протеомики, они также могут использоваться для

оценки выравниваний, полученных путём сравнения последовательностей.

1.1 Данные, получаемые путём структурного выравнивания

Результатом работы программ структурного выравнивания, как правило, является совмещение наборов координат атомов и наименьшее среднеквадратическое отклонение (RMSD) между структурами. Кроме того, могут быть рассчитаны и более сложные параметры, оценивающие структурное сходство, например, тест глобальных расстояний. RMSD указывает на степень расхождения выравниваемых структур. Структурное выравнивание может быть затруднено из-за наличия нескольких доменов в структуре выравниваемых белков, так как изменения в относительном расположении этих доменов между двумя структурами могут искусственно изменять значение RMSD. Из структурного выравнивания непосредственно вытекает соответствующее одномерное выравнивание последовательностей, кроме того, на его основании можно рассчитать долю аминокислотных остатков, идентичных между двумя белками.

1.2 Типы сравнений

Для создания структурного выравнивания и подсчёта соответствующих значений RMSD могут быть использованы как все атомы, входящие в молекулу белка, так и их подмножества. Например, атомы боковых радикалов аминокислотных остатков учитываются не всегда, и для выравнивания могут использоваться только атомы, входящие в пептидный остов молекулы. Такой вариант выбирают, если у выравниваемых структур очень разная аминокислотная последовательность и боковые радикалы различаются у большого числа остатков. По этой причине по умолчанию методы пространственного выравнивания используют только атомы остова, вовлечённые в пептидную связь. Для большего упрощения и увеличения эффективности часто используется положение только альфа-атомов углерода, так как их положение до-

статочно точно определяет положение атомов полипептидного остова. Только при выравнивании очень похожих или даже идентичных структур важно учитывать позиции атомов боковых цепей. В этом случае RMSD отражает не только схожесть конформации белкового остова, но и ротамерные состояния боковых цепей. Другие способы, позволяющие снизить шум и увеличить число правильных сопоставлений, используют разметку элементов вторичной структуры, карты нативных контактов или паттерны взаимодействия остатков, меры степени упаковки боковых цепей и меры сохранения водородных связей.

Самый простой способ сравнить две структуры не требует выравнивания самих структур, а использует выравнивание последовательностей. Оно определяет, какие пары аминокислотных остатков сопоставлены друг другу, и затем только они используются для подсчёта RMSD. Наложение структур обычно используется для сравнения нескольких конформаций одного белка (в этом случае даже не нужно выравнивать последовательности) и для оценки качества выравниваний последовательностей, если для них известны структуры. Традиционно при наложении структур используется простой метод наименьших квадратов, в котором оптимальные повороты и трансляции находят через минимизацию суммы квадратов расстояний между всеми структурами в наложении. В недавнее время подобный поиск стал более точным благодаря методам максимального правдоподобия и байесовским методам.

Алгоритмы, основанные на многомерных поворотах и модифицированных кватернионах, были разработаны для определения топологических отношений между структурами белков без построения выравнивания последовательностей. Такие алгоритмы успешно определили канонические укладки, такие как четырёхспиральный пучок. Метод SuperPose позволяет учитывать относительные вращения доменов и другие сложные моменты структурного выравнивания.

1.3 Представление структур

Для того чтобы сравнивать структуры белков, нужно представить их в пространстве, которое не зависит от координат. Это обычно достигается с помощью матрицы «последовательность против последовательности» или серии матриц, которые включают меры сравнения, относящиеся к фиксированному пространству координат, а не абсолютные расстояния. Очевидным способом подобного представления является матрица расстояний, которая представляет собой двумерную матрицу, содержащую все попарные расстояния между некоторым набором атомов в каждой структуре (например, альфа-атомами углерода). Размерность такой матрицы с увеличением числа одновременно сравниваемых структур растёт. Представив белок в виде крупных частей, таких как элементы вторичной структуры (SSEs) или другие структурные фрагменты, тоже можно получить разумное выравнивание, несмотря на потерю информации от неучтенных расстояний, так как не будет учитываться и шум от них. Таким образом, выбор способа представления белка для облегчения вычислений является критическим для разработки эффективного алгоритма выравнивания.

1.4 Вычислительная сложность

Оптимальное решение

Было показано, что оптимальное «протягивание» белковой последовательности через известную структуру и построение оптимального множественного выравнивания последовательностей являются NP-полными задачами. Однако обычная задача структурного выравнивания не является NP-полной. Строго говоря, оптимальное решение задачи структурного выравнивания белков известно только для некоторых мер сходства белковых структур – например, мер, используемых в задачах предсказания структуры белка GDT_TS и MaxSub. Такие меры могут быть оптимизированы, используя алгоритм, способный максимизировать число атомов в двух белках, которые могут быть совмещены, так как удовлетворяют предустановленному порогу на расстояние между ними. К сожалению, алгоритм оптимального выров-

нивания непрактичен, так как время его работы зависит не только от длин последовательностей, но и от геометрии выравниваемых белков.

Приближенное решение

Были разработаны и приближённые алгоритмы структурного выравнивания, работающие полиномиальное время и выдающие целое семейство «оптимальных» решений в пределах параметра приближения для заданной функции счёта. Хотя теоретически задача приближённого структурного выравнивания белков легко даётся таким алгоритмам, они всё равно являются вычислительно затратными для масштабного анализа белковых структур. Как следствие, не существует практичных алгоритмов, которые с заданной функцией счёта сходились бы к глобальному решению выравнивания. По этой причине большинство алгоритмов являются эвристическими, но всё же были разработаны практичные алгоритмы, которые гарантируют схождение хотя бы к локальной максимизации функции счёта.

1.5 Методы

Структурное выравнивание используется как при сравнении отдельных структур или их наборов, так и при создании баз данных сравнений «все-против-всех» («all-to-all»), которые отражают различия между каждой парой структур, присутствующих в Protein Data Bank (PDB). Такие базы данных обычно используются для классификации белков по их укладке.

DALI

Одним из популярных методов структурного выравнивания является DALI (англ. distance alignment matrix method — метод с использованием матрицы дистанционных выравниваний). В нём исходные структуры белков разбиваются на гексапептиды и через оценку паттернов контактов между фрагментами рассчитывается матрица расстояний. Элементы вторичной структуры, остатки которых являются соседними в последовательности, оказываются на главной диагонали матрицы; остальные диагонали матрицы отражают пространственные контакты между остатками, которые в последовательности не находятся рядом друг с другом. Если эти диагонали параллельны главной диагонали, то элементы вторичной структуры, которые они представляют,

тоже параллельны; если они, напротив, перпендикулярны ей, то их элементы вторичной структуры антипараллельны. Такое представление интенсивно работает с памятью, так как используемая матрица симметрична относительно главной диагонали (и потому избыточна).

Когда матрицы расстояний двух белков имеют одинаковые или похожие элементы примерно на одинаковых позициях, можно сказать, что белки имеют схожую укладку и их элементы вторичной структуры соединены петлями примерно одинаковой длины. Непосредственный процесс выравнивания DALI заключается в поиске схожестей матриц, построенных для двух белков; обычно это осуществляется с помощью серии перекрывающихся подматриц размера 6×6. Соответствия подматриц потом пересобираются в конечное выравнивание с помощью стандартного алгоритма максимизации счёта. Оригинальная версия DALI использует симуляцию Монте-Карло для максимизации значения пространственной схожести, являющегося функцией от расстояний между предполагаемыми соответствующими атомами. В частности, вес более отдалённых атомов внутри соответствующих элементов структуры экспоненциально занижается, чтобы уменьшить шум, вызванный подвижностью петель, искривлением спиралей и другими мелкими вариациями структур. Поскольку DALI основан на матрице расстояний типа «все-против-всех», метод может учитывать расположение элементов структур в различном порядке в двух сравниваемых последовательностях.

Метод DALI был использован для создания базы данных FSSP (англ. Families of Structurally Similar Proteins), в которой все известные структуры белков были попарно выровнены для определения их пространственного родства и классификации укладок.

DaliLite является скачиваемой программой, использующей алгоритм DALI.

Комбинаторное расширение

Метод комбинаторного расширения (англ. Combinational extension (CE)) похож на DALI тем, что тоже разбивает каждую структуру на ряд фрагментов, которые затем пытается заново собрать в полное выравнивание. Серия попарных сочетаний фрагментов, называемых AFP (англ. aligned fragment pairs — пары выровненных фрагментов), используется для задания матрицы сходства, через которую прокладывается оптимальный путь для определе-

ния конечного выравнивания. Только те AFP, которые удовлетворяют заданным критериям локального сходства, включаются в матрицу, что сокращает необходимое пространство поиска и увеличивает эффективность. Возможны разные меры сходства; в первоначальном виде метод CE использовал только структурные совмещения и расстояния между остатками, но со временем был расширен для использования локальных свойств, таких как вторичная структура, доступность растворителя, паттерны водородных связей и двугранные углы.

Путь, соответствующий выравниванию, рассчитывается как оптимальный путь через матрицу сходства с помощью линейного прохода через последовательности, расширяя выравнивание следующей возможной AFP с высоким счётом. Начальная AFP, инициирующая выравнивание, может быть выбрана в любой точке матрицы последовательностей. Далее происходит расширение на AFP, которая удовлетворяет заданному критерию на расстояние, ограничивающему размер гэпов (разрывов) в выравнивании. Размер каждой AFP и наибольшая длина гэпа являются необходимыми входными параметрами, но обычно устанавливаются равными эмпирически определённым значениям 8 и 30 соответственно. Подобно DALI или SSAP, CE использовался для создания базы данных классификации укладок на основе известных пространственных структур белков из PDB. Недавно PDB выпустил обновлённую версию CE, которая может определять циклические перестановки в структуре белков.

SSAP

Метод SSAP (англ. Sequential Structure Alignment Program) использует двойное динамическое программирование для построения структурного выравнивания, основанного на векторах «от атома к атому» в пространстве структур. Вместо альфа-атомов углерода, обычно используемых в структурных выравниваниях, SSAP задаёт свои вектора из бета-атомов для всех аминокислотных остатков за исключением глицина. Таким образом, этот метод учитывает положение ротамера каждого остатка, также как и их положение в остове. Сначала SSAP для каждого белка строит серию векторов расстояний между каждым остатком и его ближайшим, но не смежным в последовательности соседом. После этого конструируется ряд матриц, содержащих разницу векторов между соседями для каждой пары остатков, для которых

строились вектора. Для каждой получившейся матрицы с помощью динамического программирования определяется ряд оптимальных локальных выравниваний. Затем полученные выравнивания складываются в обобщённую матрицу, к которой снова применяется динамическое программирование для определения полного структурного выравнивания. Изначально SSAP создавал только попарные выравнивания, но в дальнейшем был расширен и для создания множественных выравниваний. Он был применён для выравнивания типа «все-против-всех» для создания иерархической системы классификации укладок, известной как CATH, которая используется в базе данных CATH Protein Structure Classification.

Глава 2

Методы и результаты

2.1 Предварительные сведения и определения

Задача выравнивания структуры пары белков может быть сформулирована следующим образом: "Учитывая два белка a и b и расстояние среза $\sigma > 0$, найти жёсткое преобразование t и соответствие остаток-остаток (выравнивание), которое максимизирует количество пар остатков a и $t(b)$ на расстоянии $\leq \sigma$". Мы называем t – σ-оптимальное преобразование для a и b. Отметим, что, не теряя общности, можно предположить, что белок a удерживается фиксированным, в то время как белок b трансформируется.

Для того, чтобы точно сформулировать вышеуказанную проблему, нам нужны некоторые определения.

Определение Белок – это последовательность точек в трёхмерном пространстве:

$$a = (a_1, a_2, \ldots, a_n), a_i \in \mathbb{R}^3, i = \overline{1, n}$$

Во многих приложениях, a_i представляет собой CA-остаток. **Определение** Выравнивание белков $a = (a_1, a_2, \ldots, a_n)$ и $b = (b_1, b_2, \ldots, b_m)$ – это последовательность пар точек из a и b:

$$S(a, b) = ((a_{i_1}, b_{j_1}), \ldots, (a_{i_k}, b_{j_k}))$$

где $1 \leq i_1 < \cdots < i_k \leq n$ и $1 \leq j_1 < \cdots < j_k \leq m$. **Определение** σ-оптимальное выравнивание белков $a = (a_2, a_2, \ldots, a_n)$ и $b = (b_1, b_2, \ldots, b_m)$,

обозначается $S(a, b, \sigma)$ это выравнивание a и b, которое максимизирует число выровненных точек a и b на расстоянии σ.

В приведённом выше определении мы предполагаем, что белки a и b зафиксированы в пространстве. $S(a, b, \sigma)$ относятся к оптимальному спариванию аминокислотных букв буще изменения структурной суперпозиции. Хорошо известно, что для любых двух фиксированных в пространстве белков a и b длины n и любого $\sigma > 0$ выравнивание $S(a, b, \sigma)$ может быть вычислено за время $O(n^2)$ с использованием динамического программирования. Теперь мы будем использовать $|S(a, b, \sigma)|$ для обозначения количества пар точек в $S(a, b, \sigma)$ на расстоянии $\leq \sigma$

Определение σ-оптимальная трансформация для a и b, обозначается t^σ, – это жёсткая трансформация, которая максимизирует $|S(a, t(b), \sigma)|$ в множестве всех жёстких трансформаций. Другими словами:

$$t^\sigma = \arg\max_t |S(a, t(b), \sigma)|$$

Легко увидеть, что CA $\leq \sigma$ в точность $|S(a, t^\sigma(b), \sigma)|$, где t^σ – это σ-оптимальная трансформация a и b.

Определение Трансформация t удовлетворяет:

$$|S(a, t(b), \sigma + \varepsilon)| \geq |S(a, t^\sigma(b), \sigma)|$$

Это называется (σ, ε)-трансформация a и b, обозначается t^σ_ε.

Нужно отметить, что t^σ_ε – это любая трансформация t с таким свойством, что пары точек в структурах a и $t(b)$, которые могут быть совмещены на расстоянии $\leq \sigma + \varepsilon$ не меньше, чем количество таких пар в структурах a и $t^\sigma(b)$.

Так же напомним, что любая ориентация, сохраняющая жёсткое преобразование t является композиции вращения и переноса $t = t_{tr} \circ t_{rot}$. Любая такая трансформация может быть представлена как точка в шестимерном пространстве:

$$t = (\alpha, \beta, \gamma, u, v, w) \in \mathbb{R}^6$$

где α и γ – угла поворота вдоль оси z, β – это поворот вдоль оси x. а также u, v, w – это перенос вдоль оси x, y, z.

2.2 Приближённое решение

Без потери общности можно предположить, что оба белка a и b имеют центр масс в начале. Таким образом, суперпозиция, выравнивающая центр масс белков a и b, будет первой (среди многих) суперпозиций, проверенной данным методом. Рассматриваемый алгоритм нахождения t_ε^σ, называемый ε-оптимальным, основан на непрерывности жёстких преобразований. Другими словами, небольшое изменение любого из шести аргументов $T = (\alpha, \beta, \gamma, u, v, w)$ приводит к небольшому изменению пространственного положения белка B. Например, если b вращается вокруг оси x на небольшой угол $\delta > 0$, то расстояние пройденное любой точкой $p(x, y, z) \in b$ будет равно:

$$d = \sqrt{2(y^2 + z^2)(1 - \cos\delta)} \leq \sqrt{2} R_b \sin\delta \leq \sqrt{2} R_b \delta$$

где R_b – это радиус минимальной сферы содержащей в себе b.

Каждый элемент группы поворотов $R \in SO(3)$ может быть представлен как:

$$R = R_z(\alpha) R_x(\beta) R_z(\gamma)$$

где $(\alpha, \beta, \gamma) \in [0, 2\pi) \times [0, \pi] \times [0, 2\pi)$ и R_x и R_y обозначающие поворот вокруг осей x и y определяется следующим образом:

$$R_x(\phi) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{vmatrix}$$

$$R_z(\phi) = \begin{vmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Данное представление уникально за исключением случая $\beta = 0$ и $\beta = \pi$. Угла α, β, γ называются Эйлеровыми углами поворота R.

2.3 Алгоритма нахождения почти оптимального решения

Несложно увидеть, что кандидатами для t_ε^σ являются только те преобразование, которые не удаляют белок b далеко от a, т.е. преобразования $t = (\alpha, \beta, \gamma, u, v, w) \in \mathbb{R}^6$ из замкнутого интервала:

$$I = [0, 2\pi] \times [0, \pi] \times [0, 2\pi] \times [-M_x, M_x] \times [-M_y, M_y] \times [-M_z, M_z]$$

где

$$M_x = \frac{m_x^a + m_x^b}{2}, M_y = \frac{m_y^a + m_y^a}{2}, M_z = \frac{m_z^a + m_z^b}{2}$$

и $m_x^a, m_y^a, m_z^a, m_x^b, m_y^b, m_z^b$ –размер наименьшего интервала $B(a)$ и $B(b)$ в $\mathbb{R}^3$ содержащие a и b.

Для дальнейшего уменьшения размера пространства определим конечную ε-сеть преобразований из интервала I, полученную разбиением на небольшие шестимерные ячейки. Точки ε-сети являются вершинами этих ячеек.

$$d_1 = d_2 = d_3 = \frac{\varepsilon}{3\sqrt{2}R_b} \tag{2.1}$$

$$d_4 = d_5 = d_6 = \frac{\varepsilon}{\sqrt{3}} \tag{2.2}$$

Заметим, что для нахождения t_ε^σ достаточно проверить преобразования из ε-сети. Это происходит потому, что для каждого $t = (\alpha, \beta, \gamma, u, v, w) \in I$, существует такая формула преобразования, что:

$$|\alpha - \overline{\alpha}| \leq \frac{d_1}{2}, |\beta - \overline{\beta}| \leq \frac{d_2}{2}, |\gamma - \overline{\gamma}| \leq \frac{d_3}{2} \tag{2.3}$$

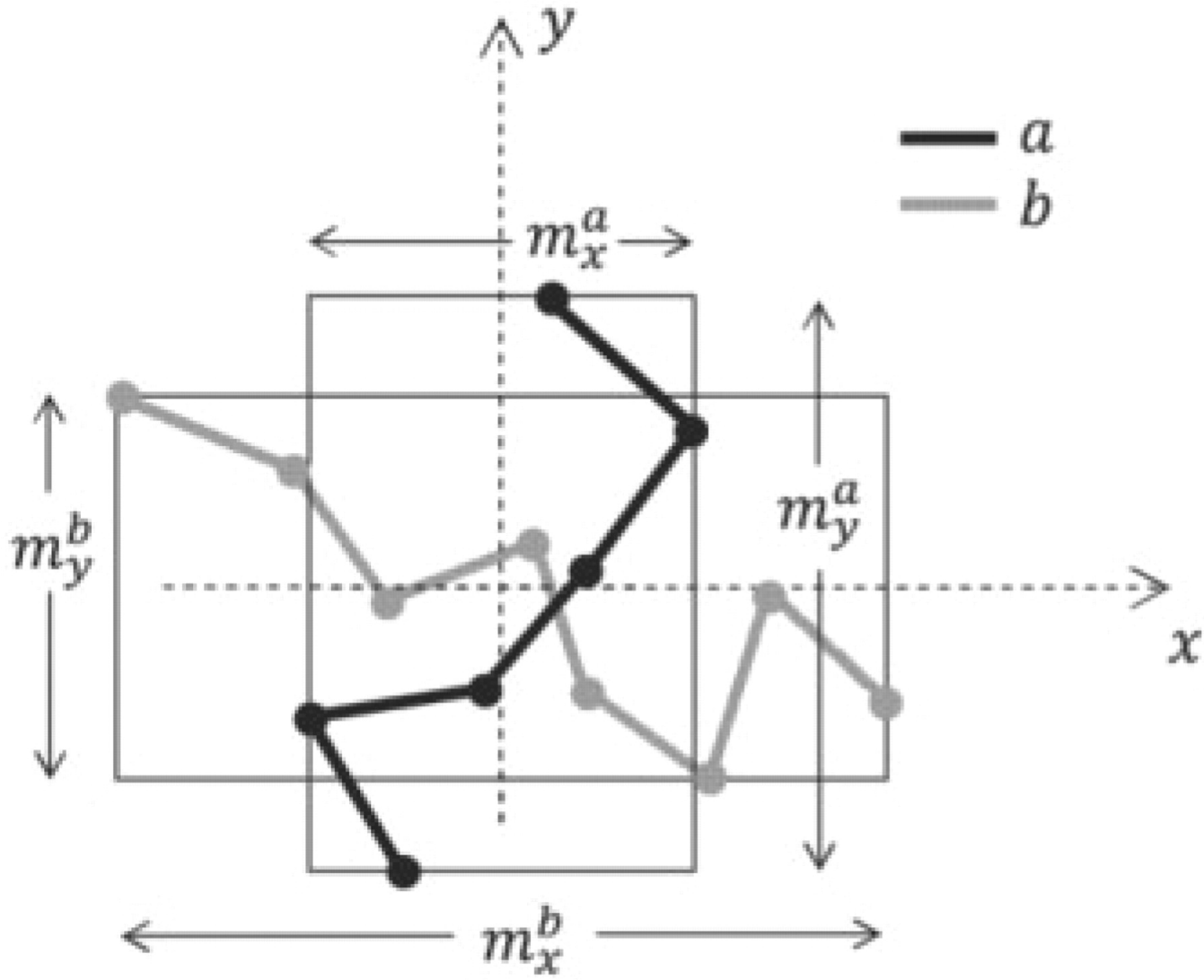
$$|u - \overline{u}| \leq \frac{d_4}{2}, |v - \overline{v}| \leq \frac{d_5}{2}, |w - \overline{w}| \leq \frac{d_6}{2} \tag{2.4}$$

и следовательно

$$\|t(b_i) - \overline{t}(b_i)\| \leq \varepsilon, \forall b_i \in b \tag{2.5}$$

где $\|t(b_i) - \overline{t}(b_i)\|$ обозначает Евклидово расстояние между $t(b_i)$ и $\overline{t}(b_i)$. В частности, если t – это σ-оптимальное преобразование t^σ, тогда $\overline{t}$ – это (σ, ε)-оптимальное преобразование t_ε^σ.

Рис. 2.1:



Докажем это утверждение:

$$\triangleright$$

$$\|t(b_i) - \overline{t}(b_i)\| = \|t_{tr}(t_{rot}(b_i)) - \overline{t_{tr}}(\overline{t_{rot}}(b_i))\| =$$

$$= \|t_{tr}(t_{rot}(b_i)) - t_{tr}(\overline{t_{rot}}(b_i)) + t_{tr}(\overline{t_{rot}}(b_i)) - \overline{t_{tr}}(\overline{t_{rot}}(b_i))\| \leq$$

$$\leq \|t_{tr}(t_{rot}(b_i)) - t_{tr}(\overline{t_{rot}}(b_i))\| + \|t_{tr}(\overline{t_{rot}}(b_i)) - \overline{t_{tr}}(\overline{t_{rot}}(b_i))\| =$$

$$= \|t_{rot}(b_i) - \overline{t_{rot}}(b_i)\| + \|\overline{t_{rot}}(b_i) - \overline{t_{rot}}(b_i)\| \leq$$

$$\leq \sqrt{2}R_b(|\alpha - \overline{\alpha}| + |\beta - \overline{\beta}| + |\gamma - \overline{\gamma}|) + \sqrt{(u - \overline{u})^2 + (v - \overline{(v)})^2 + (w - \overline{w})^2} \leq$$

$$\leq \sqrt{2}R_b\left(\frac{d_1}{2} + \frac{d_2}{2} + \frac{d_3}{2}\right) + \sqrt{\frac{d_4^2}{4} + \frac{d_5^2}{4} + \frac{d_6^2}{4}} = \varepsilon$$

$$\triangleleft$$

Псевдокод данного алгоритма:

Перенесём центры масс белков a и b в начало координат.
Вычислим M_x, M_y, M_z, R_b.
$r_{\alpha,\gamma} \leftarrow \left[\frac{6\sqrt{2}\pi R_b}{\varepsilon}\right]$
$r_\beta \leftarrow \left[\frac{3\sqrt{2}\pi R_b}{\varepsilon}\right]$
$s_x \leftarrow \left[1 + \frac{2\sqrt{3}M_x}{\varepsilon}\right]$
$s_y \leftarrow \left[1 + \frac{2\sqrt{3}M_y}{\varepsilon}\right]$
$s_z \leftarrow \left[1 + \frac{2\sqrt{3}M_z}{\varepsilon}\right]$
$d_r \leftarrow \frac{\varepsilon}{3\sqrt{2}R_b}$
$d_t \leftarrow \frac{\varepsilon}{\sqrt{3}}$
$bestScore \leftarrow 0$
for $0 \leq i_1 \leq r_{\alpha,\gamma}, 0 \leq i_2 \leq r_\beta, 0 \leq i_3 \leq r_{\alpha,\gamma}$ **do**
    for $0 \leq i_4 \leq s_x, 0 \leq i_5 \leq s_y, 0 \leq i_6 \leq s_z$ **do**
        $t \leftarrow (i_1 d_r, i_2 d_r, i_3 d_r, -M_x + i_4 d_t, -M_y + i_5 d_t, -M_z + i_6 d_t)$
        if $|S(a, t(b), \sigma + \varepsilon)| > bestScore$ **then**
            $bestScore \leftarrow |S(a, t(b), \sigma + \varepsilon)|$
            $t_{best} \leftarrow t$
        end if
    end for

19: **end for**

20: **return** t_{best}

2.4 Время работы

Для пары белков длины n, худший слушай работы данного алгоритма происходит, когда радиус ограничивающей сферы белка b линеен относительно n, то есть $R_b = O(n)$. В этом случае общее число преобразований, проверенных данным алгоритмом, равно $\mathbf{NET}(\varepsilon) = O\left(\frac{n^6}{\varepsilon^6}\right)$. Для каждого такого преобразования оптимальное соответствие может быть вычислено с использованием процедуры динамического программирования $O(n^2)$, что приводит к наихудшему времени работы $O\left(\frac{n^8}{\epsilon^6}\right)$. Однако общая стоимость данного алгоритма, как правило, лучше на практике, так как объём белка пропорционально масштабируется с количество остатков. Например, если b – глобулярный белок, тогда $R_b = O(n^{\frac{1}{3}})$, и таким образом время работы данного алгоритма равно только $O\left(\frac{n^4}{\varepsilon^6}\right)$. Для сравнения алгоритм Колодного и Линеала для оптимизации класса оценочных функций, удовлетворяющих условию Липшица, равна $O\left(\frac{n^{10}}{\varepsilon^6}\right)$ для глобулярных и $O\left(\frac{n^{12}}{\varepsilon^6}\right)$ для неглобулярных белков.

2.5 Качество решения

Качество решения t_ε^σ представленного ε-оптимальным алгоритмом определяется как разность между оценкой оптимального решения t^σ и оценкой t_ε^σ:

$$Err(t_\varepsilon^\sigma) = |S(a, t^\sigma(b), \sigma)| - (S(a, t_\varepsilon^\sigma(b), \sigma)| \tag{2.6}$$

Пока $Err(t_\varepsilon^\sigma)$ не может вычислен в пределах временного окна ε-оптимального алгоритма (так как не известен t^σ), верхняя граница $Err(t_\varepsilon^\sigma)$:

$$MaxErr(t_\varepsilon^\sigma) = |S(a, t_\varepsilon^\sigma(b), \sigma + \varepsilon)| - |S(a, t_\varepsilon^\sigma(b), \sigma)| \tag{2.7}$$

Данное выражение может вычислено за счёт маленькой модификации ε-оптимального алгоритма, без увеличения асимптотической сложности.

2.6 Оптимальное решение

Процедура нахождения оптимального решения базируется на том наблюдении, что $f(\sigma) = |S(a, t^{\sigma}(b), \sigma)|$ это функция, которую можно разбить на некоторое конечное число значений $\sigma_1, \ldots, \sigma_k$. Если $\sigma > 0$ – некоторое действительное число отличное от $\sigma_i, \forall i \in [1, k]$, тогда для достаточно малого $\varepsilon > 0$ получаем $f(\sigma + \varepsilon) = f(\sigma - \varepsilon)$.

$$f(\sigma+\varepsilon) = |S(a, t^{\sigma+\varepsilon}, \sigma+\varepsilon)| \geq |S(a, t_{\varepsilon}^{\sigma}(b), \sigma+\varepsilon)| \geq$$
$$\geq |S(a, t^{\sigma}(b), \sigma)| \geq |S(a, t_{\varepsilon}^{\sigma-\varepsilon}(b), \sigma)| \geq$$
$$\geq |S(a, t^{\sigma-\varepsilon(b)}, \sigma-\varepsilon)| = f(\sigma-\varepsilon)$$

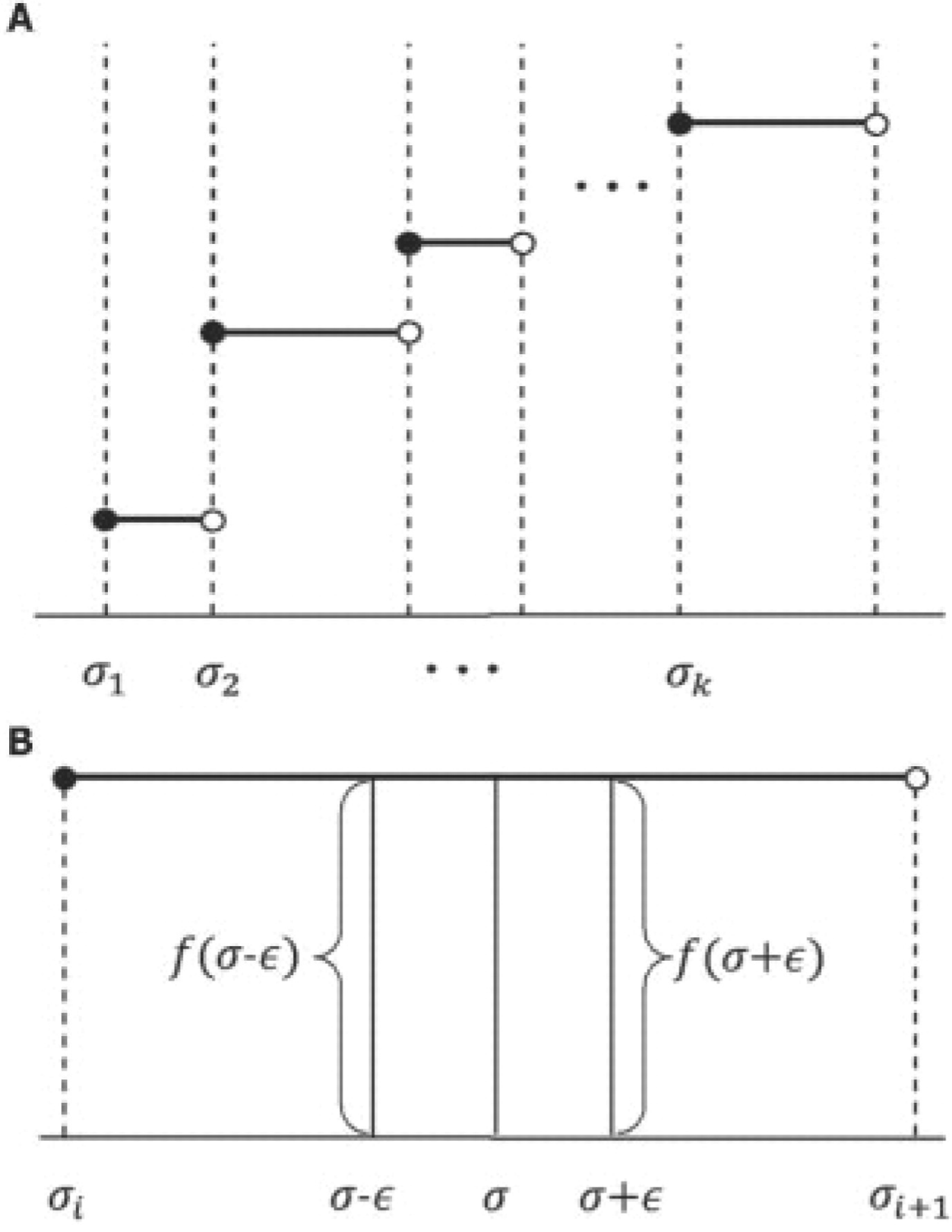
Отсюда следует, что для каждого такого ε существует преобразование $t_{\varepsilon}^{\sigma-\varepsilon}$, которое является оптимальным.

Опишем полученный алгоритм в виде псевдокода.

OPTIMAL(a, b, σ). Отметим, что OPTIMAL возвращает оптимальное решение с вероятностью 1, то есть в случаях, когда σ не совпадает ни с одним из фиксированных значений $\sigma_1, \ldots, \sigma_k$. Однако число операций, выполненных OPTIMAL, нельзя оценить заранее, так как время её выполнения зависит от разности между σ и ближайшим σ_i (которая зависит от внутренней геометрии входных белков a и b).

1: $\varepsilon \leftarrow 1$
2: $t_{\varepsilon}^{\sigma} \leftarrow \text{EPSILON} - \text{OPTIMAL}(a, b, \sigma, \varepsilon)$
3: $t_e ps^{\sigma-\varepsilon} \leftarrow \text{EPSILON} - \text{OPTIMAL}(a, b, \sigma-\varepsilon, \varepsilon)$
4: $\varepsilon \leftarrow \frac{\varepsilon}{2}$
5: **while** $|S(a, t_{\varepsilon}^{\sigma}(b), \sigma+\varepsilon)| - |S(a, t_{\varepsilon}^{\sigma-\varepsilon}(b), \sigma) > 0$ **do**
6: $\quad t_{\varepsilon}^{\sigma} \leftarrow \text{EPSILON} - \text{OPTIMAL}(a, b, \sigma, \varepsilon)$
7: $\quad t_e ps^{\sigma-\varepsilon} \leftarrow \text{EPSILON} - \text{OPTIMAL}(a, b, \sigma-\varepsilon, \varepsilon)$
8: $\quad \varepsilon \leftarrow \frac{\varepsilon}{2}$
9: **end while**
10: **return** $t_{\varepsilon}^{\sigma-\varepsilon}$

Рис. 2.2:


A
σ_1
σ_2
σ_k
B
$f(\sigma-\epsilon)$
$f(\sigma+\epsilon)$
σ_i
$\sigma-\epsilon$
σ
$\sigma+\epsilon$
σ_{i+1}

2.7 Практическое применение

Помимо относительно малого времени работы, ε-оптимальный алгоритм поддаётся параллельным вычисления, так как NET (ε) может быть разделена и процедура поиска будет проводиться одновременно на нескольких подмножествах NET(ε). Для оценки потенциальных преимуществ параллельных реализаций данного алгоритма была разработана более быстрая эвристическая версия алгоритма называемая MAX-PAIRS.

Для эффективности MAX-PAIRS исследуют только небольшое подмножество NET(ε), состоящее только из тех преобразований из NET(ε), которые близки к некоторым "начальным"преобразованиям, обладающим высокой оценкой. Предполагается, что оптимальное преобразование находится не далеко от преобразования с достаточно высокой оценкой, полученного некоторым быстрым и достаточно точным эвристическим методом.

Для вычисления каждого начального преобразования MAX-PAIRS применят хорошо известный итеративный алгоритм расширения выравнивания до $S = S(a, t(b), \sigma)$, где t-преобразование, минимизирующее RMSD между короткими отрезками k последовательных остатков в a и b (по умолчанию $k = 5$). На каждой итерации алгоритма расширения белки накладываются, чтобы минимизировать RMSD между выровненными остатками, и новое выравнивание вычисляется динамическим программированием. Вся процедура повторяется до тех пор, пока длина $|S(a, t(b)), \sigma)|$ остаётся неизменной между двумя последовательными итерациями.

После генерации всех начальных преобразований с высокой оценкой, MAX-PAIRS "уточняют"их по одному исследую ближайшие трансформации из NET(ε). Более конкретно (и предполагая, что начальным трансформации уже применены к b), алгоритм выбирает три пары выровненных точек

$$\{(a_{i_k}, b_{i_k})\}, k = \overline{1,3}$$

из $S(a, b, \sigma)$, а затем ищет NET(ε), сохраняя точки a_{i_k} и b_{i_k} связанными, то есть лежащими на расстоянии не более σ. Изучая только преобразования τ такие, что $\|a_{i_k} - \tau(b_{i_k})\| \leq \sigma, \forall k \in \{1, 2, 3\}$. MAX-PAIRS значительно уменьшает размер пространства поиска, что приводит к повышению эффективности.

Опишем процедуру уточнения более подробнее. Без потери общности полагаем, что точка b_{i_1} – это точка начала координат, точка b_{i_2} – лежит в положительной части прямой y, а точка b_{i_3} лежит на плоскости yz.

Для уменьшения размера пространства поиска, мы можем исключить из него все преобразования $t = (\alpha, \beta, \gamma, u, v, w)$ такие, что $|u| > 2\sigma$, или $|v| > 2\sigma$, или $|w| > 2\sigma$, потому что каждое такое преобразование ломает связь между a_{i_1} и b_{i_1}:

$$\|t(b_{i_1}) - a_{i_1}\| = \|t(b_{i_1}) - b_{i_1} + b_{i_1} - a_{i_1}\| \geq$$
$$\geq \|t(b_{i_1}) - b_{i_1}\| - \|b_{i_1} - a_{i_1}\| > 2\sigma - \sigma = \sigma$$

Также можно избавиться от определённых комбинаций углов α и γ, то есть от преобразований $t = (\alpha, \beta, \gamma, u, v, w)$ таких, что:

$$\cos(\alpha + \gamma) < 1 - \frac{8\sigma^2}{\|b_{i_2}\|^2}$$

и любой из $\alpha, gamma \in [0, \frac{\pi}{2}] \cup [\frac{3\pi}{2}, 2\pi]$ или $\alpha, \gamma \in [\frac{\pi}{2}, \frac{3\pi}{2}]$. Для всех таких преобразований:

$$\|t_{rot}(b_{i_2} - b_{i_2}\| = \sqrt{2}\|b_{i_2}\|\sqrt{1 + \sin\alpha\sin\gamma - \cos\alpha\cos\beta\cos\gamma} \geq$$
$$\geq \sqrt{2}\|b_{i_2}\|\sqrt{1 - \cos(\alpha + \gamma)} > 4\sigma$$

и поэтому:

$$\|t(b_{i_2} - a_{i_2}\| = \|t_{rot}(b_{i_2} - b_{i_2} + t_{tr}(t_{rot}(b_{i_2})) - t_{rot}(b_{i_2}) + b_{i_2} - a_{i_2}\| \geq$$
$$\geq \|t_{rot}(b_{i_2}) - b_{i_2}\| - \|t_{tr}(t_{rot}(b_{i_2})) - t_{rot}(b_{i_2})\| - \|b_{i_2} - a_{i_2}\| >$$
$$> 4\sigma - 2\sigma - \sigma = \sigma$$

(полагается, что пространство было уменьшено до преобразований удовлетворяющих условию $\|t_{tr}(t_{rot}(b_{i_2})) - t_{rot}(b_{i_2})\| < 2\sigma$).

Аналогичная аргументация может быть использована чтобы показать, что если $\alpha \in [0, \frac{\pi}{2}] \cup [\frac{3\pi}{2}, 2\pi]$ и $\gamma \in [\frac{\pi}{2}, \frac{3\pi}{2}]$, или наоборот, то можно уменьшить пространство преобразований удалив преобразования $t = (\alpha, \beta, \gamma, u, v, w)$, такие что:

$$\cos(\alpha - \gamma) > \frac{8\sigma^2}{\|b_{i_2}\|^2} - 1$$

Рис. 2.3:

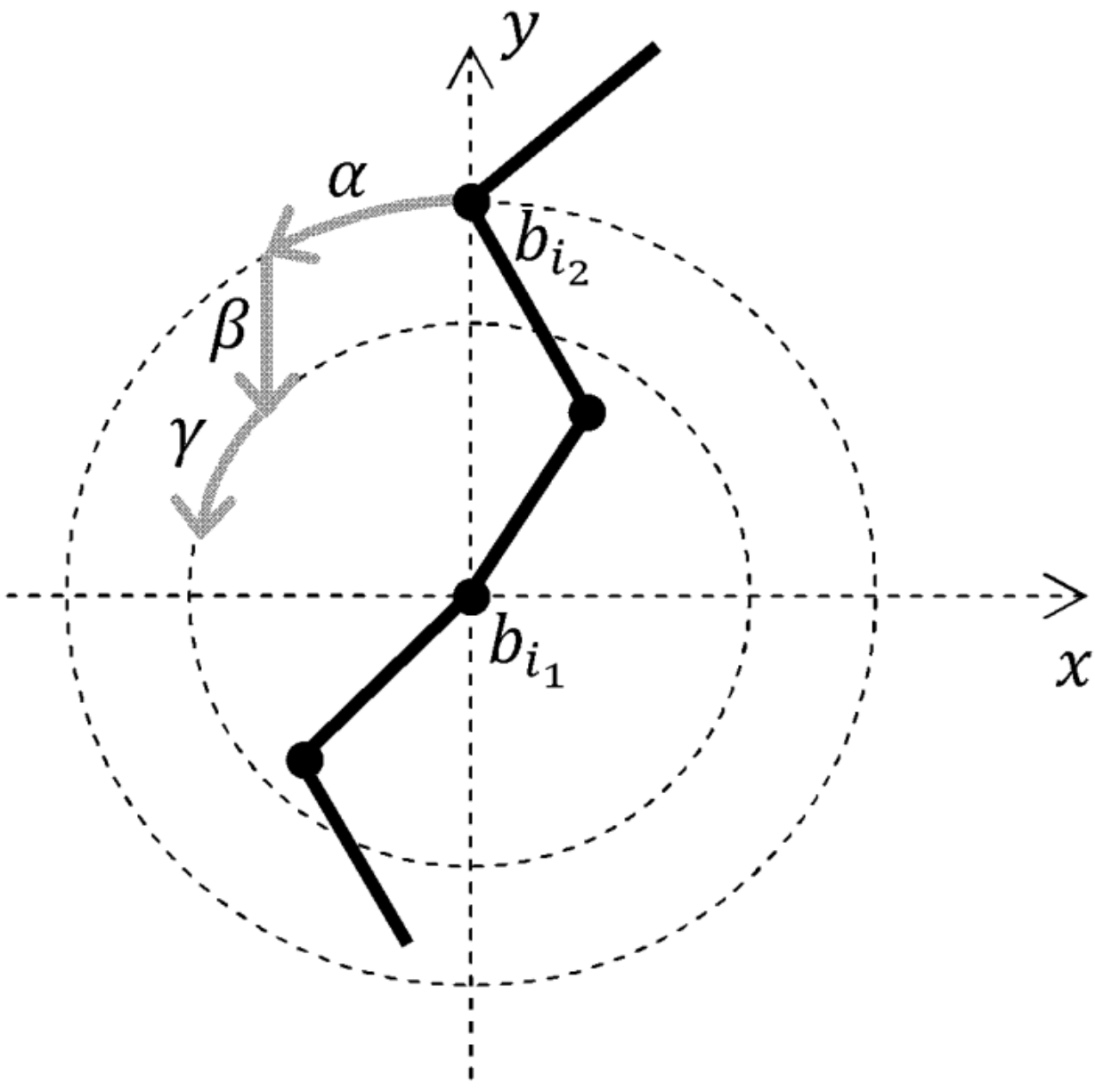


Таблица 2.1: Общее число пар в тестовом наборе, которые могут быть наложены на расстояние не превосходящем 3 Å

	MAX-PAIRS	LGA	TM-alogn	Mammoth	Mustang
Family	4689	4585	4460	4264	4231
Superfamily	4378	4247	4140	3713	3319
Flod	2870	2720	2634	2100	1834

Для того, чтобы уменьшить пространство преобразований ещё больше, отметим, что если r – это точка на оси z такая что
$r\| = \|b_{i_3}\|$, тогда каждое преобразование удовлетворяющее следующему неравенству:

$$\cos\beta < 1 - \frac{2(2\sigma + d)^2}{\|b_{i_3}\|^2}$$

ломает связь между a_{i_3} и b_{i_3}, где $d = \|r - b_{i_3}\|$.

И, наконец, не сложно заметить, что наилучшее уменьшение пространства преобразований достигается, когда выбираются точки $\{b_{i_k}\}_{k=1}^{3}$ так, чтобы максимизировать $\|b_{i_2}\|$ и $\|b_{i_3}\|$, а также минимизировать d.

2.8 Результаты работы

Производительность MAX-PAIRS была протестирована на репрезентативном наборе белковых цепочек, отобранных из базы данных SCOP. Набор для тестирования содержит 195 пар белков связанных на различных условиях согласно структурной классификации SCOP: 57 family пар, 75 superfamily пар, и 63 fold-пары.

Для эффектиности параметр точности ε MAX-PAIRS устанавливается равным 1 (уменьшение ε даёт более точную, но менее эффективную процедуру). Оценки для MAMMOTH и MUSTANG, представленные в таблицах, показаны только для справки, так как в отличие от MAX-PAIRS и LGA, которые максимизируют функцию $CA \leq \sigma$, эти программы стремятся оптимизировать другую целевую функцию.

Рис. 2.4:

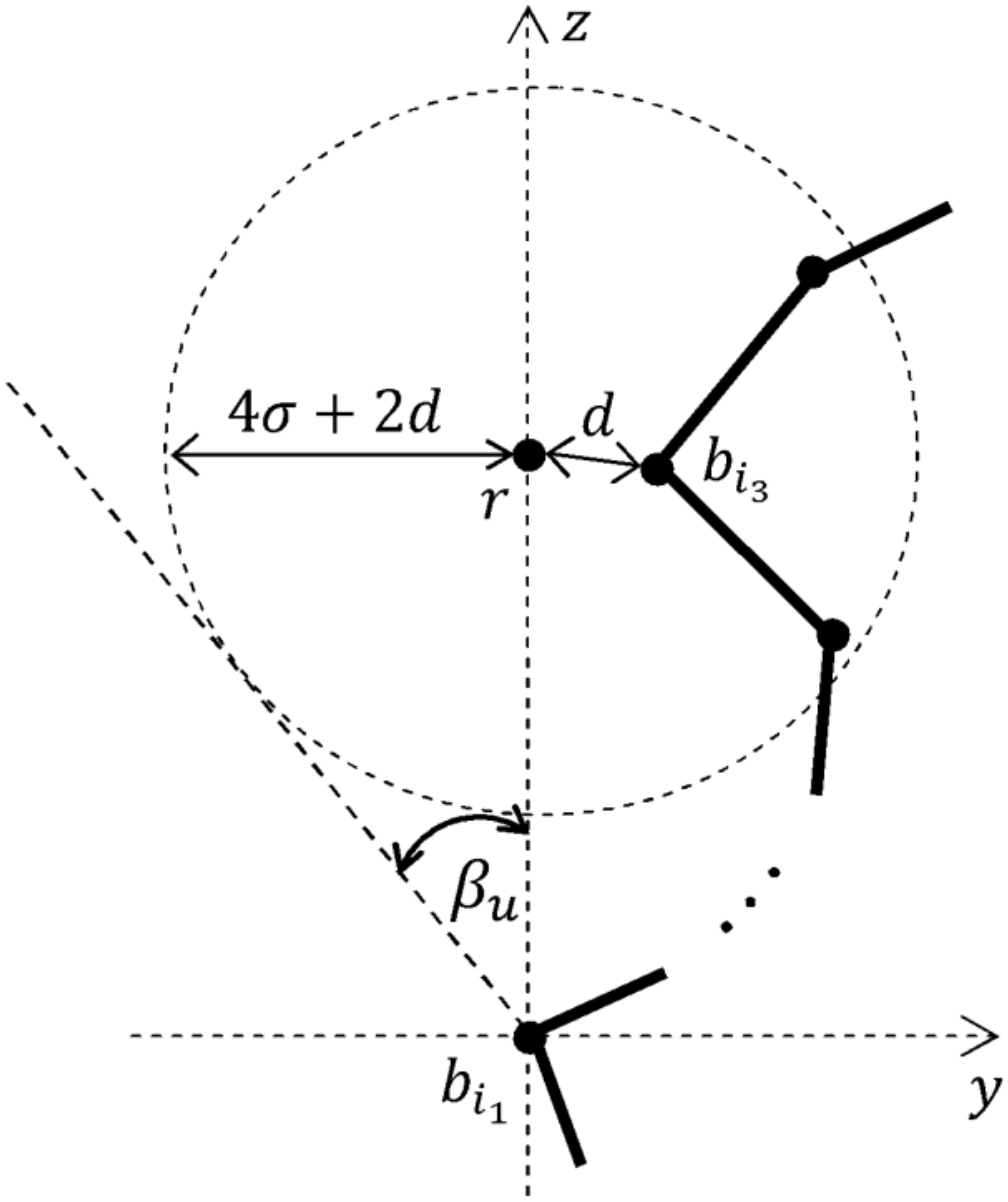
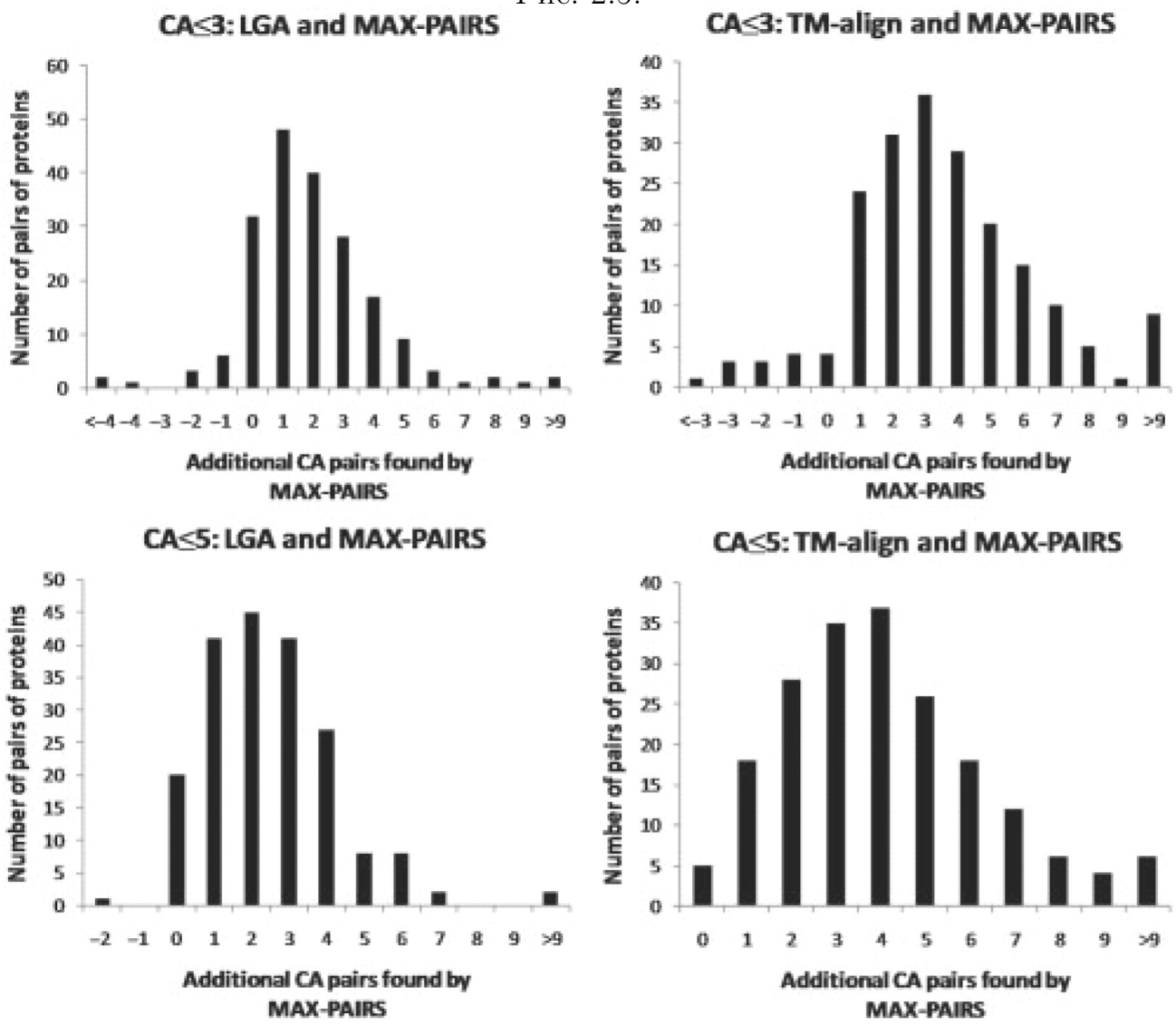


Таблица 2.2: Общее число пар в тестовом наборе, которые могут быть наложены на расстояние не превосходящем 5 Å

	MAX-PAIRS	LGA	TM-alogn	Mammoth	Mustang
Family	5261	5130	5059	5019	4983
Superfamily	5240	5033	4928	4702	4532
Flod	3575	3409	3279	2842	2816

Рис. 2.5:



Как видно из таблиц 1 и 2, даже с $\varepsilon = 1$ MAX-PAIRS лучше, по сравнению с другими методами, показываем себя на всех категориях SCOP на обоих расстояних (3Åи 5Å)

На следующем графике можно увидеть распределение добавленного значения MAX-PAIRS, измерено дополнительное число $CA \leq \sigma$ обнаруженных данным алгоритмом. Как видно из левых хвостов на графиках, на некоторых парах белков MAX-PAIRS работает хуже, чем LGA и TM-align. Это не удивительно так как все эти алгоритмы использует различные наборы преобразований для поиска лучшей суперпозиции.

В таблице 3 покажем эффективность MAX-PAIRS как функции зависящей от ε она множестве пар структур из описанного выше датасета. Анализ был произведён на компьюетере с Intel CPU производительностью 2.2 GHz с

Рис. 2.6:

Таблица 2.3: Время работы в зависимости от ε

ε	$CA \leq 3$	Время работы одной пары (секунд)
1.0	11937	6608
1.5	11862	713
2.0	11789	140
2.5	11711	46
3.0	11602	19
3.5	11566	9

операционной системой Linux. Результаты собранные в данной таблице предполагают что, хотя LGA и TM-align намного более эффективные программы по сравнению с MAX-PAIRS однако точность MAX-PAIRS превосходит точности этих алгоритмов для всех протестированных значений ε.

Многие методы попарного структурного выравнивания белков, включая методы, обсуждаемые в этой работе, используют ключевую процедуру для вычисления суперпозиций, которая максимизирует $CA \leq \sigma$ между парой белковых структур. Вполне разумно ожидать, что улучшение этой суперпозиции повышает точность выравнивания структуры белка. Для оценки степени этого улучшения, используется Sisyphus бенчмарк для точности выравнивания Набор тестов Sisyphus содержит 125 созданных вручную структурных выров-

ниваний для пар белков с нетривиальным структурными отношениями. Эти выравнивания могут быть использованы (как золотые стандарты) для оценки точности метода выравнивания структуры белка. Для того, чтобы сравнить точность выравнивания алгоритмов в данной работе с точностью методов , ранее протестированных в Sisyphus бенчмарке, использовались только подмножество набора теста Sisyphus содержащего 106 одиночных цепных белков.

Чтобы проверить полезность алгоритмов оптимизации $CA \leq \sigma$, мы модифицировали метод TM-align, заменив исходные суперпозиции TM-align суперпозициями, порождёнными программой MAX-PAIRS. Модифицированная TM-align, названная MP-TM-align, использует TM-align функцию оценивания (TM-score) для вычисления оптимального структурного выравнивания белков совмещённых MAX-PAIRS программой. Как видно из таблицы 4, не только программа MP-TM-align превосходит исходный метод TM-align для каждого сдвига допуска, но и точность этого простого гибридного метода сопоставима с точностью методов проверенных Rocha и коллегами.

Интересно отметить, что, согласно исследованию Rocha, наиболее точные методы структурного выравнивания, такие как Matt, PPM и ProtDeform, рассматривают белки, как гибкие объекты. Эти методы достигают высокой точность выравнивания, применяя последовательность различных жёстких преобразований на разных участках, а не одно глобальное жёсткое преобразование. С другой стороны, результаты данного исследования показывают, что всё ещё могут быть разработаны высокоточные методы, которые опираются на одно жёсткое преобразование для оценки сходства белковых структур.

Ещё большее повышение точности TM-align может быть достигнуто за счёт использования информации о типе остатков в процессе выравнивания. Комбинирование мер на основе расстояния с оценками мутации остатков является стандартным методом, используемым во многих методах выравнивания структуры, таких как CE. Как видно из таблицы 4, вариант MP-TM-align метода, названный MP-TM-align+, которое основано на совмещённой функции оценки, определённой как сумма TM-score и BLOSUM62 score даёт самое точное выравнивание согласно Sisyphus benchmark.

При более подробном рассмотрении результатов, обобщённых в таблице 4, и результатов, полученных Rocha и другими, выявлена существенная разни-

Таблица 2.4: Согласование с эталонными выравниваниями для 6 различных сдвигов

	0	1	2	3	4	5
FLEXPROT	0.449	0.672	0.707	0.725	0.742	0.747
MATRAS	0.776	0.806	0.828	0.836	0.847	0.847
PD	0.791	0.849	0.858	0.868	0.881	0.882
PPM	0.782	0.813	0.823	0.833	0.843	0.844
RASH	0.688	0.793	0.812	0.840	0.854	0.855
SSAP	0.750	0.786	0.797	0.804	0.808	0.811
VOROLIGN	0.722	0.765	0.790	0.808	0.826	0.830
DALI	0.800	0.830	0.845	0.851	0.859	0.860
MATT	0.829	0.866	0.889	0.904	0.915	0.917
LGA	0.765	0.820	0.831	0.839	0.847	0.849
TM-align	0.762	0.815	0.823	0.834	0.841	0.844
MP-TM-align	0.809	0.861	0.875	0.884	0.896	0.896
MP-TM-align+	0.830	0.867	0.881	0.887	0.897	0.898

ца TM-align в этих двух исследованиях. Это различие объясняется тем, что в двух экспериментах использовались разные версии TM-align. Более конкретно, программа TM-align, протестированная в данном тесте, выпущена 14 марта 2009 года и на 4% более точна, чем старая программа, оценённая в тесте Rocha.

Выравнивание белков в наборе тестов Sisyphus, созданное LGA, TM-align, MP-TM-align и MP-TM-align+ может быть загружено отсюда:
`http://bioinformatics.cs.uni.edu/opt_align.html`.

Выравнивание сгенерированное десятью другими методами может быть скачано отсюда:
`http://dmi.uib.es/people/jairo/bio/ProtDeform`.

Также осталось много работы по ускорению MAX-PAIRS, требуется сделать его практичным для крупномасштабного анализа структуры белков. Однако даде в нынешнем видел MAX-PAIRS может быть полезен при оценке эффективности методов прогнозирования структуры белка. Например точность MAX-PAIRS на 3,6% выше точности программы LGA, официально используемой на двухгодичном конкурсе CASP. Это значительное преимущество метода MAX-PAIRS,учитывая, что разница в баллах GDT_TS между первым и вторым методами ранжирования в CASP7, измеренная LGA, составляет всего 2.6% (3.5% в CASP8).

Заключение

Выводы о подобии двух белков зачастую получают на основе их структурного сходства. Однако из-за бесконечного (несчётного) пространства всех возможных пространственных конфигураций, нахождение суперпозиции для пары белков является очень трудоёмкой задачей.

В этой работе показано, что проблему приближённого структурного выравнивания можно трактовать различными метриками структурного выравнивания, такими как GDT, AL0 и MaxSub. Хотя описанный в данной статье алгоритм для почти оптимального решения потребляет большое вычислительное время, время работы можно легко уменьшить с помощью параллельных реализаций. Дополнительным преимуществом алгоритма для приближённого решения является то, что он обеспечивает меру качества решения, которая сигнализирует, является ли возвращённая суперпозиция, по сути, оптимальной суперпозицией.

Также в данной работе представлена процедура, способная найти оптимальную суперпозицию любых двух белков для всех, кроме конечного числа порогов, расстояний. Однако такая программа для поиска оптимального решения слишком медленная для практического применения.

Список литературы

1 Andreeva A , et al. SISYPHUS—structural alignments for proteins with non-trivial relationships , Nucleic Acids Res. , 2007 , vol. 35 (pg. D253 -D259)

2 Alexandrov NN , et al. Common spatial arrangements of backbone fragments in homologous and nonhomologous proteins , J. Mol. Biol. , 1992 , vol. 225 (pg. 5 -9)

3 Caprara A , et al. 1001 optimal PDB structure alignments: integer programming methods for finding the maximumcontact map overlap , J. Comput. Biol. , 2004 , vol. 11 (pg. 27 -52)

4 Csaba G , et al. Protein structure alignment considering phenotypic plasticity , Bioinformatics , 2008 , vol. 24 (pg. i98 -i104)

5 Eidhammer I , et al. Structure comparison and structure patterns , J. Comput. Biol. , 2000 , vol. 7 (pg. 685 -716)

6 Fischer D , et al. CAFASP3: the third critical assessment of fully automated structure prediction methods , Proteins , 2003 , vol. 53 Suppl. 6 (pg. 503-516)

7 Ginalski K , et al. Practical lessons from protein structure prediction , Nucleic Acids Res. , 2005 , vol. 33 (pg. 1874 -1891)

8 Goldman D , et al. Algorithmic aspects of protein structure similarity , Proceedings of the 40th Annual Symposium on Foundations of Computer Science , 1999 Washington, DC, USA IEEE Computer Science (pg. 512 -522)

9 Goldsmith-Fischman S , Honig B . Structural genomics: computational methods for structure analysis , Prot. Sci. , 2003 , vol. 12 (pg. 1813 -1821)

10 Henikoff S , Henikoff JG . Amino acid substitution matrices from protein blocks , Proc. Natl Acad. Sci. USA , 1992 , vol. 89 (pg. 10915 -10919)

11 Hao MH , et al. Effects of compact volume and chain stiffness on the conformations of native proteins , Proc. Natl Acad. Sci. USA , 1992 , vol. 89 (pg. 6614 -6618)

12 Holm L , Sander C . Protein structure comparison by alignment of distance matrices , J. Mol. Biol. , 1993 , vol. 233 (pg. 123 -138)

13 Kabsch W . solution for the best rotation to relate two sets of vectors , Acta Crystallographica , 1976 , vol. 32 (pg. 922 -923)

14 Kolodny R , Linial N . Approximate protein structural alignment in polynomial time , Proc. Natl Acad. Sci. USA , 2003 , vol. 101 (pg. 12201 -12206)