**УДК 004.942**

# ОПТИМИЗАЦИЯ МОДЕЛИРОВАНИЯ БЕЛКОВЫХ ВЗАИМОДЕЙСТВИЙ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ С ПРЕДОСТАВЛЕНИЕМ ДОСТУПА К АЛГОРИТМУ ЧЕРЕЗ ВЕБ-ИНТЕРФЕЙС[1]

***К.В. Романенков, А.Н. Сальников***

В работе представлены параллельные версии последовательной программы создания молекулярных интерфейсов с применением технологии OpenMP и MPI. Обе версии показали достаточную масштабируемость и лучшие временные показатели по сравнению с последовательной версией при запуске на одном процессоре. Моделирование белкового интерфейса для некоторых соединений занимает более двадцати часов счета на нескольких сотнях процессоров, поэтому для задач моделирования белковых соединений с большим количеством позиций важно наличие недетерминированных алгоритмов, позволяющих за приемлемое время получать биологически корректный результат. Выбор стохастических алгоритмов оправдал себя: и метод Монте-Карло, и алгоритм пчелиного поиска нашли пространственное расположение молекулы, соотвествующее минимальному энергетическому состоянию. Предоставление доступа к реализации алгоритма по веб-интерфейсу отвечает современным тенденциям к перемещению вычислений на сторону сервера и позволяет широкому кругу специалистов использовать вычислительные мощности, предоставляемые Московским государственным университетом, а с учетом расширения сферы применимости задач молекулярного моделирования наличие открытого веб-интерфейса, предоставляющего удаленный доступ к вычислительным кластерам, является достаточно важной задачей.

*Ключевые слова: биинформатика, многопроцессорные системы, создание молекулярных интерфейсов, параллельные алгоритмы, стохастические алгоритмы.*

## Введение

Существуют два основных подхода к моделированию белков: основанные на информации о пептидной последовательности и основанные на информации о структуре молекулы. Классическим представителем задачи первого типа является проблема фолдинга. Надо заметить, что на сегодняшний день подходы для решения задач этой категории далеко не очевидны, достаточно упомянуть, что рекордные длины искусственно сгенерированных белков с заданными свойствами трехмерной структуры составляют около трех сотен [1], при том, что, к примеру, в молекуле титина камбаловидной мышцы человека содержится более 25000 аминокислотных остатков.

Задача вычислительного моделирования белковых соединений относится ко второму типу задач. Одним из важнейших направлений в этой области считается создание молекулярных интерфейсов [2], позволяющее, в частности, предсказывать, какие аминокислотные остатки надо заменить в соединениях, не взаимодействующих в природе, но обладающих заданными свойствами, чтобы добиться их взаимодействия.

Существуют различные подходы для решения задач этого типа, из последовательных детерминированных версий можно выделить метод ветвей и границ [2] и алгоритм A* [3]. Общей проблемой этих методов являются большие временные затраты при обработке значительных объемов данных. В качестве методов, использующих распределенные вычислению,

[1]Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: поиск новых решений – 2012».

можно выделить реализации, построенные на GRID-системах [4, 5], в которых заранее неизвестно, какое количество ресурсов каждый компьютер в сети сможет выделить для решения задачи, и реализацию, использующую технологию MPI для организации итеративного поиска решения [6], при этом большее число процессоров позволяет получать более точный результат и не ведет к ускорению работы.

Статья организована следующим образом. В разделе 1 описывается предметная область и обсуждаются особенности вычислительного моделирования белковых соединений. В разделе 2 содержится краткое описание используемых в исследовании алгоритмов и данных. Раздел 3 описывает модификацию исходной программы с целью созданию параллельных версий и использования стохастических алгоритмов. В разделе 4 описывается система *Aligner*, в которую была интегрирована созданная реализация. В разделе 5 приведены результаты вычислительных экспериментов и результаты работы стохастических алгоритмов. В заключении оценивается выигрыш от использования выбранных методов оптимизации и описываются направления будущих исследований.

## 1. Принципы компьютерного моделирования белковых соединений

Основной целью компьютерного моделирования белковых соединений является выбор аминокислот в воспроизводимой структуре, минимизирующих общую энергию системы [7]. Считается, что в каждом взаимодействующем домене (структурно обособленной единице) есть фиксированное число позиций, в которые возможно подставлять различные аминокислоты с целью получения минимального энергетического состояния (GMEC – Global Minimum Energy Conformation). Гибкость аминокислот аппроксимируется ротамерами – конформационными изомерами, отличающимися от других конформеров углами поворота. Принята модель, в которой известно число углов вращения, принимающих конечное число значений. Данные о структурах ротамеров собраны в специальные библиотеки, где каждой совокупности значений углов вращения сопоставлен конформер. В рамках такой модели общая энергия системы складывается из трех основных компонент:

1) энергии остова, остающаяся неизменной при поиске GMEC и поэтому не участвующая в оптимизации;
2) энергии взаимодействия ротамера с остовом молекулы;
3) энергии взаимодействия двух определенных ротамеров между собой.

Общая энергия вычисляется по формуле

$$\varepsilon(C) = E_1 + \sum_{i=1}^{n} E_2(C_i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E_3(C_i, C_j), \tag{1}$$

где $i$ – ротамер в $i$-й позиции.

Вычислительное моделирование белковых соединений имеет следующие ключевые особенности:

1. Количество позиций, участвующих в образовании интерфейса не очень велико (менее 100).
2. В каждой позиции моделируется мутация природного аминокислотного остатка, что влечет за собой рассмотрение большего количества ротамеров по сравнению с традиционными задачами структурного моделирования (синтез гомологов, при котором

рассматривается большее количество позиций, но с уже известными природными ротамерами, у которых необходимо рассмотреть лишь возможные пространственные изомеры).

3. Так как функция энергии допускает некоторую погрешность, в качестве результата работы алгоритма нужно получить список конформаций с минимальными энергиями, для которых производится честное моделирование, гораздо более требовательное к временным ресурсам.
4. Неоднородная энергетическая поверхность (это одна из сложностей применения алгоритмов самосогласованного поля (SCMF) [8]).
5. Проблема GMEC – NP-полная [8].

## 2. Параметры исследования

### 2.1. Описание используемых методов оптимизации

В качестве стохастических алгоритмов (то есть выдающих результат, зависящий не только от входных данных, но и от датчика случайных чисел), были выбраны MC/Q и SBC, так как они обладают достаточно потенциалом для выхода из локальным минимумов, которыми изобилует задача создания молекулярных интерфейсов. В качестве базового метода дискретной оптимизации использовался DEE.

*DEE (Dead End Elimination).* Простейший метод дискретной оптимизации, позволяет отсекать заведомо неперспективные ротамеры, основываясь на их вкладе в общую энергию [8].

*MC/Q(Monte Carlo / Quenching).* Несмотря на то, что метод был разработан в середине 1950-х годов, успешно применяться в задачах моделирования белковых соединений он начал лишь в 1990-х годах [8]. Суть метода заключается в следующем: вначале последовательность ротамеров инициализируется случайным образом. Затем в случайной позиции происходит подстановка другого ротамера, причем замены на ротамеры разных аминокислот, включая ту, что стояла в выбранной позиции, равновероятны. После подстановки происходит вычисление энергии конформации, и если она меньше предыдущего значения, то замена принимается. Если новая энергия больше, то замещение подтверждается с вероятностью Больцмана, $k$ – постоянная Больцмана. Вероятность Больцмана задает вероятность подстановки менее энергетически выгодного ротамера в определенную позицию и расчитывается по формуле:

$$P = \mathrm{e}^{\frac{-(E_{new}-E_{old})}{k \cdot T}}. \qquad (2)$$

Величина $T$ в данном случае исполняет роль температуры, позволяя избегать локальных минимумов. После завершения поиска возможен переход к фазе отжига. Для каждой позиции, выбранной в случайном порядке, перебираются все ротамеры аминокислотного остатка, который был найден в ходе работы метода Монте-Карло. Если энергия новой конформации будет меньше, то происходит замещение ротамера. Этот этап позволяет убедиться, что в полученной структуры отсутствуют подстановки отдельных конформеров, минимизирующих общую энергию. Отжиг имеет малую временную сложность, однако может серьезно улучшить найденное решение.

*Алгоритм пчелиного поиска (Simulated Bee Colony).* Алгоритм, появившийся в 2005 году, хорошо зарекомендовал себя как в задачах непрерывной, так и дискретной оптимизации [9]. Кратко его можно описать следующим образом: в начале работы случайным

образом выбирается $m$ решений, каждое из которых представляет собой пчелу разведчика. Затем циклически лучшие $n$ решений исследуются более тщательно: в зависимости от того является ли точка «элитной» или просто выбранной, в её окрестности исследуется $s_e$ или $s_p$ случайных решений, а остальные $(m - n)$ решений заменяются на случайные точки из пространства решений. Завершение алгоритма происходит либо в результате достижения определенной точности, либо после исчерпания числа итераций.

## 2.2. Описание входных данных для исследования взаимодействия структур

Для исследования были использованы два типа белковых структур, отражающих реальную задачу моделирования белковых интерфейсов. Постановка задачи для определенных белков сформелирована институтом физико-химической биологии имени А.Н. Белозерского. Были исследованы: белковый комплекс LAGLIDADG эндонуклеаз и белковый комплекс антитело – антиген.

Эндонуклеазы — белки из семейства нуклеаз, узнающие длинные последовательности ДНК и вносящие в найденный фрагмент двунитевой разрыв. Они часто используются в генной инженерии для создания рекомбинантных ДНК, которые затем могут вводиться в клетки других организмов. Эндонуклеаза семейства LAGLIDADG состоит из двух доменов, и задача, связанная с ними, состояла в комбинировании отдельных доменов из различных белков с целью получения нуклеаз с новой специфичностью.

Антитела представляют собой специальные белки системы иммунитета. Они связываются с большой силой взаимодействия с антигеном (с высокой степенью диссоциации) – характерной частью патогена, например белковый капсид вируса, — и либо нейтрализуют антиген, либо маркируют его для других клеток иммунитета. Особенность строения антител заключается в том, что основной каркас молекулы не меняется, различаются только вариабельные петли, непосредственно отвечающие за взаимодействие с антигеном. Основным способом получения антител на сегодняшний день является иммунизация животных, это достаточно дорогая и неудобная процедура, поэтому становится актуальной задача оптимизации как фармокинетических свойств антитела (растворимость, стабильность), так и его аффинности (прочность связи с антигеном). Можно выделить два направления развития в искусственном подборе антител:

1) оптимизация комплекса антитело – антиген средствами компьютерного моделирования для повышения аффинности антитела к антигену;
2) дизайн новых антител к указанным антигенам «de novo», имитирующий селективный отбор иммунной системы.

Относительно недавно были открыты и сейчас активно исследуются наноантитела [10, 11]: молекулы, сходные по структуре с антителами, но меньшего размера (в частности, меньше объем и количество вариабельных цепей), что дает им серьезные преимущества в синтезе лекарств перед антителами обычного размера. В качестве входных данных для задачи был предложен комплекс наноантитела с лизоцимом, который исполнял роль антигена. Стоит отметить вычислительную сложность этой задачи: исходя из того, что количество операций, необходимых для обработки 1 конформации, зависит от квадрата числа позиций, и зная общее количество комбинаций, получаемое перемножением числа всевозможных ротамеров в позициях, получаем, для моделирования наноантитела даже в 6 позициях (хотя может быть задействовано более 20) требует $6 \cdot 6 \cdot 191 \cdot 308 \cdot 190 \cdot 106 \cdot 196 \cdot 326$

операций с плавающей точкой, то есть требуется компьютер с производительностью более 2,7 ПФлопс.

## 3. Программа FitProt

Коллективом института им. А.Н. Белозерского в распоряжение авторов была предоставлена последовательная программа *FitProt* [12], написанная на языке программирования Python, которая осуществляла полный перебор конформаций для указанных пользователем позиций, предварительно применяя к ним фильтр DEE (см. пункт 2.1). На вход программе подаются два файла: со структурой соединения в формате PDB и с описанием в каких позициях и у каких ротамеров надо моделировать мутацию остатка. Второй файл имеет текстовый формат.

С целью ускорения работы последовательной реализации и для обеспечения возможности распараллеливания кода в *FitProt* были внесены некоторые изменения. Для хранения списка лучших конформаций, минимизирующих энергию системы, была реализована куча (heap), что было обусловлено меньшими временными затратами на добавление нового элемента и поддержания структуры кучи по сравнению с другими способами представления данных, а также скоростью сортировки. Выполнена реализация программы, использующая массив энергий, сгенерированный программой *FitProt* с использованием фильтра DEE, на языке C/C++, которая затем была распараллелена с использованием технологий OpenMP и MPI.

Авторами была произведена модификация программы *FitProt* так, чтобы ее было возможно запускать на многопроцессорных системах как с общей памятью, так и кластерной архитектуры. В OpenMP реализации все данные об энергиях ротамеров хранятся в общей памяти, а в MPI распределяются по процессорам. В обоих случаях параллельно рассматриваются различные конформации; затем из списков, сформированных различными MPI-процессорами, и OpenMP-нитями, строится список минимальных энергий.

Для дальнейшего сокращения времени поиска решения ко входным данным были применены модифицированные алгоритмы MC/Q и GMEC. Несмотря на то, что оба алгоритма предназначены для выдачи единственного решения, при создании реализации они были модифицированы для получения списка минимальных конформаций. Очевидно, что большое значение на результаты их работы оказывает точный выбор параметров, подходящих под решаемую задачу.

Алгоритм MC/Q реализован с применением технологии MPI, в котором между процессами распределяется пространство поиска, то есть большее количество процессоров не ведет к ускорению программы, зато теоретически может обеспечивать нахождение лучшего решения за то же время. Модификация алгоритма для получения списка минимальных конформаций состоит в следующем: согласно каноническому этапу MC на каждом процессоре ищется оптимальная конформация, после этого на этапе отжига при рассмотрении всех вариантов ротамеров аминокислотного остатка, стоящего в выбранной позиции, все полученные конформации добавляются в список решений. С увеличением количества позиций средняя энергия решения имеет тенденцию к снижению, что говорит о хорошем потенциале для масштабирования. Разделение пространства поиска между процессорами имеет целью оптимизацию найденных решений помимо GMEC, когда на каком-то процессоре фиксируется локальный минимум и, возможно, улучшается на шаге отжига.

Реализован последовательный алгоритм пчелиного поиска, в качестве окрестности элитных точек рассматривались все варианты ротамеров аминокислотного остатка в случайно выбранной позиции, в качестве окрестности выбранных точек брались все ротамеры в случайной позиции с равной вероятностью. Количество элитных, выбранных и остальных точек соотносится как 10:25:100 соответственно, такие цифры, согласно эмпирическим исследованиям, позволяют получать оптимальный результат.

## 4. Описание системы Aligner

С целью обеспечения простоты использования создаваемого программного кода было решено интегрировать созданную параллельную версию в систему *Aligner* [13]. Изначально система *Aligner* создавалась как интернет-сервис для построения множественного выравнивания последовательностей на кластере и включала в себя систему авторизации, базу данных, возможность загрузки пользовательских данных, поддержку уведомлений о статусе заданий по электронной почте [14]. Преимущество *Aligner* над остальными веб-интерфейсами к многомашинным комплексам заключается в отсутствии у пользователя необходимости регистрации на вычислительных кластерах, к которым предоставляет доступ *Aligner*. Вместо этого достаточно завести учетную запись в системе, позволяющую ставить на счет задачи на всех суперкомпьютерах, связанных с *Aligner*. Структура системы *Aligner* представлена на рис. 1.

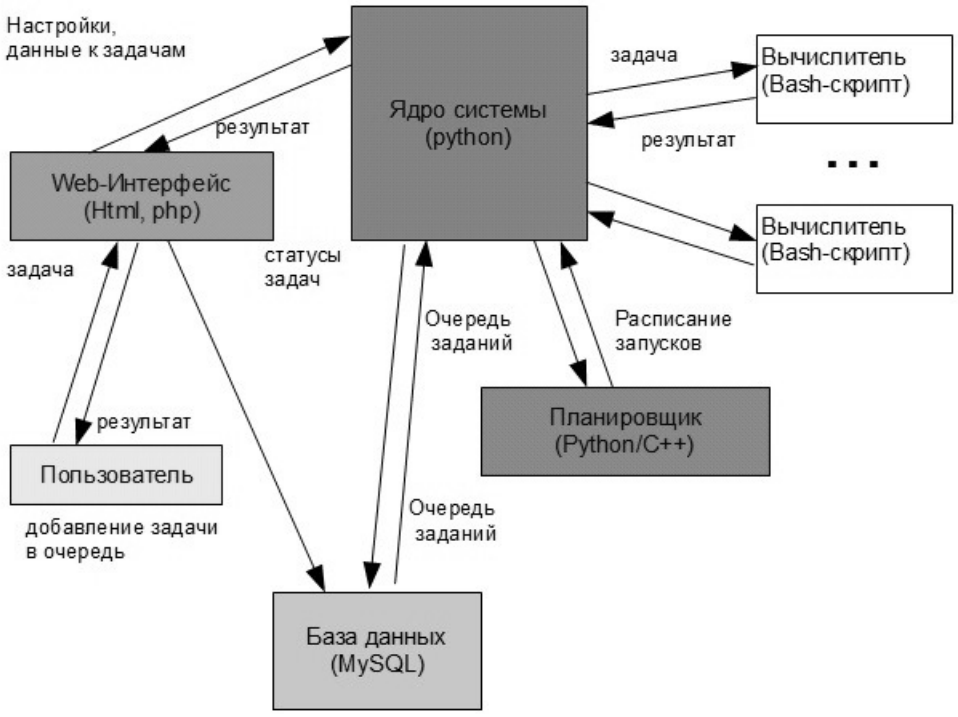


**Рис. 1.** Структура системы Aligner

В список алгоритмов, доступных для удаленного запуска на кластере, был добавлен FitProt. При выборе этого алгоритма, веб-интерфейс предлагает пользователю загрузить файл структуры в формате PDB и текстового файла, указывающего в каких позициях следует провести инжиниринг. После успешного считывания задания из базы данных происходит вызов исходного последовательного решения, которое с применением фильтра DEE

генерирует файл значений собственных и парных энергий, который отправляется на кластер, где ставится на счет программа, использующая его как параметр. В случае успешного завершения задания пользователю предоставляется возможность просмотреть и скачать выходные данные. Поведение программной системы в этом случае проиллюстрировано на рис. 2.


**Рис. 2.** Путь данных, при удаленном запуске заданий

## 5. Результаты

Для исследования эффективности параллельной реализации и как средство для удаленного запуска программы через веб-интерфейс была выбрана система Regatta, обладающая 16 процессорами Power4, располагающими 64 Гб общей памяти, и доступная из сети МГУ. Для расчетов молекулярного интерфейса наноантитела использовался комплекс «ЧЕБЫШЕВ», работающий на процессорах Intel Xeon E5472 3.0 ГГц и имеющий пиковую производительность 60 Тфлопс.

Табл. 1 содержит времена работы (в секундах) MPI версии программы, производившей моделирование молекулярного интерфейса для 3 позиций эндонуклеазы, и OpenMP версии программы, моделировавшей молекулярный интерфейс для 4 позиций эндонуклеазы. На рис. 3, приведенном ниже, указаны ускорения параллельных версий детерминированного алгоритма, исходя из данных табл. 1.

Во всех случаях стохастические алгоритмы нашли GMEC (конформация, соответствующая минимальной энергии, занимает первое место в списке, отсортированном по возрастанию энергии) и выдали несколько результатов из первой десятки минимальных конформаций. Лучшие результаты алгоритма пчелиного поиска объясняются способом получения списка оптимальной последовательности ротамеров: в MC/Q результат формируется в процессе отжига из лучшего найденного решения на этапе MC, когда перебираются конфор-

**Таблица 1**

Время работы параллельных версий программы (с)

| **Число процессоров** | **Время работы MPI версии** | **Время работы OpenMP версии** |
|---|---|---|
| 1 | 6,86 | 312 |
| 2 | 3,43 | 159 |
| 4 | 1,73 | 80 |
| 8 | 0,92 | 43 |
| 16 | 0,56 | 22 |

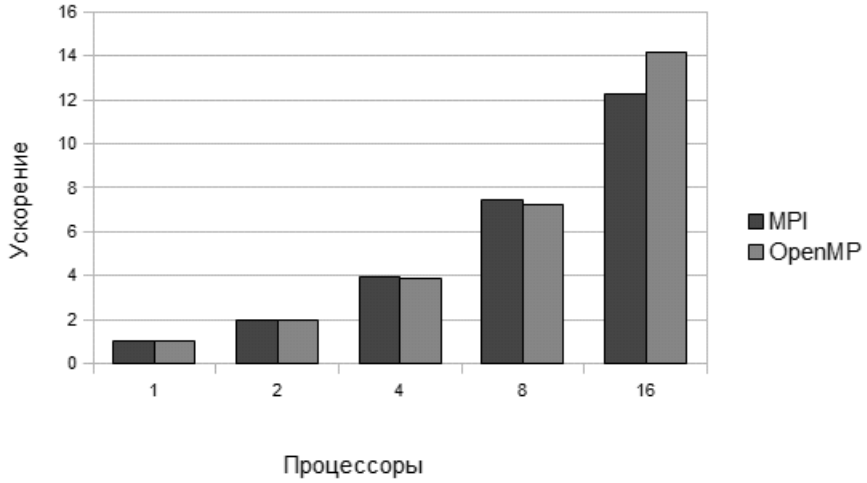


**Рис. 3.** Ускорение для параллельных версий детерминированного алгоритма

меры аминокислот, стоящих в решении, что ограничивает вариативность ответа. Алгоритм пчелиного поиска лишен этого недостатка, но существует опасность схождения всех решений к одному единственному глобальному минимуму при слишком долгом времени работы. В табл. 2 и табл. 3 приведены результаты работы алгоритмов, число найденных конформа-

**Таблица 2**

Результаты работы алгоритма MC/Q

| **Структура** | **Число позиций** | **Найдено конформаций** |
|---|---|---|
| Эндонуклеазы | 3 | 3 |
| Эндонуклеазы | 4 | 1 |
| Эндонуклеазы, увеличенное число ротамеров | 4 | 1 |
| Наноантитело в комплексе с лизоцимом | 4 | 4 |
| Наноантитело в комплексе с лизоцимом | 5 | 4 |

**Таблица 3**

Результаты работы алгоритма пчелиного поиска

| Структура | Число позиций | Найдено конформаций |
|---|---|---|
| Эндонуклеазы | 3 | 3 |
| Эндонуклеазы | 4 | 4 |
| Эндонуклеазы, увеличенное число ротамеров | 4 | 3 |
| Наноантитело в комплексе с лизоцимом | 4 | 3 |
| Наноантитело в комплексе с лизоцимом | 5 | 4 |

ций указано без учета GMEC, который был обнаружен во всех случаях. Число итераций при работе алгоритма пчелиного поиска с 3 позициями эндонуклеазы уменьшено на порядок, чтобы избежать схождения всех решений к GMEC.

## Заключение

В работе были представлены параллельные версии последовательной программы создания молекулярных интерфейсов с применением технологии OpenMP и MPI. Обе версии показали достаточную масштабируемость и лучшие временные показатели по сравнению с последовательной версией. Выбор стохастических алгоритмов оправдал себя: и Монте-Карло, и алгоритм пчелиного поиска продемонстрировали высокую способность к выходу из локальных минимумов. Стоит упомянуть, что моделировании белкового интерфейса для наноантитела в 6 позициях занимает более 20 часов счета на нескольких сотнях процессоров, поэтому для задач моделирования белковых соединений с большим количеством позиций важно наличие недетерминированных алгоритмов, позволяющих за приемлемое время получать биологически корректный результат. Предоставление доступа к алгоритму по веб-интерфейсу отвечает современным тенденциям к перемещению вычислений на сторону сервера. Интеграция в систему *Aligner* позволяет широкому кругу специалистов использовать вычислительные мощности, предоставляемые Университетом, а с учетом расширения сферы применимости задач молекулярного моделирования, наличие открытого веб-интерфейса, предоставляющего удаленный доступ к вычислительным кластерам, является достаточно важной задачей.

В качестве направлений будущих исследований можно указать поиск оптимальных коэффициентов в функции энергии, например, с помощью методов машинного обучения и исследование применимости графических процессоров для ускорения работы программы.

## Литература

1. Fortenberry, C. Exploring Symmetry as an Avenue to the Computational Design of Large Protein Domains / C. Fortenberry, E.A. Bowman, W. Proffitt, B. Dorr, S. Combs, J. Harp, L. Mizoue, J. Meiler // Journal of the American Chemical Society. — 2011. — Vol. 133, No. 45. — P. 18026–18029.

2. Wernisch, L. Automatic protein design with all atom force-fields by exact and heuristic optimization / L. Wernisch, S. Hery, S.J. Wodak // Journal of Molecular Biology. — 2000. — Vol. 301, No. 3. — P. 713–736.

3. Leach, A.R. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm / A.R. Leach, A.P. Lemon // Proteins: Structure, Function, and Bioinformatics. — 1998. — Vol. 33, No. 2. — P. 227–239.

4. Tantar, A.A. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid / A.A. Tantar, N. Melab, E.G. Talbi, B. Parent, D. Horvath // Future Generation Comp. Syst. — 2009. — Vol. 23, No. 3. — P. 398–409.

5. Pitman, D.J. Improving computational efficiency and tractability of protein design using a piecemeal approach. A strategy for parallel and distributed protein design. — Bioinformatics. — 2013. / D.J. Pitman, C.D. Schenkelberg, Y.M. Huang, F.D. Teets, D. DiTursi, C. Bystroff. URL: http://dx.doi.org/10.1093/bioinformatics/btt735 (дата обращения: 30.12.2013).

6. Molto, G. Protein design based on parallel dimensional reduction / G. Molto, M. Suarez, P. Tortosa, J.M. Alonso, V. Hernandez, A. Jaramillo // Journal of Chemical Information and Modeling. — 2009. — Vol. 49, No. 5. — P. 1261–1271.

7. Street, A.G. Computational protein design / A.G. Street, S.L. Mayo. URL: http://dx.doi.org/10.1016/S0969-2126(99)80062-8 (дата обращения: 24.02.2013).

8. Voigt, C.A. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design / C.A. Voigt, D.B. Gordon, S.L. Mayo // Journal of Molecular Biology. — 2000. — Vol. 299, No. 3. — P. 789–803.

9. Pham, D.T. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems / D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi // 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS'2006) — 2006. — P. 454–461.

10. Hamers-Casterman, C. Naturally occurring antibodies devoid of light chains / C. Hamers-Casterman, T. Atarhouch, S. Muyldermans, G. Robinson, C. Hamers, E.B. Songa, N. Bendahman, R. Hamers // Nature. — 1993. — Vol. 363. — P. 446–448.

11. Revets, H. Nanobodies as novel agents for cancer therapy / H. Revets, P. De Baetselier, S. Muyldermans // Expert Opinion on Biological Therapy. — 2005. — Vol. 5, No. 1. — P. 111–124.

12. Grishin, A. Bioinformatics analysis of LAGLIDADG homing endonucleases for construction of enzymes with changed DNA recognition specificity / A. Grishin, I. Fonfara, W. Wende, D. Alexeyevsky, A. Alexeyevsky, S. Spirin, O. Zanegina, A. Karyagina // 4-th Moscow Conference on Computational Molecular Biology. — Moscow, MSU, 2009. — P. 123.

13. Князев, Н. А. Система справедливого планирования и унифицированного запуска задач пользователя на суперкомпьютерах / Н.А. Князев, А.Н. Сальников // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.). — Челябинск: Издательский центр ЮУрГУ, 2010. — С. 665–666.

14. Сальников, А.Н. Интернет-сервис для построения множественного выравнивания последовательностей на многопроцессорных системах, созданный на основе data-flow модификации алгоритма MUSCLE / А.Н. Сальников // Параллельные вычислительные технологии (ПАВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта – 3 апреля 2009 г.). — Челябинск: издательский центр ЮурГУ, 2009. — С. 680–687.

Романенков Кирилл Владимирович, аспирант кафедры суперкомпьютеров и квантовой информатики факультета Вычислительной математики и кибернетики, Московский государственный университет (Москва, Российская Федерация), kromanenkov2@yandex.ru.

Сальников Алексей Николаевич, к.ф.-м.н., старший научный сотрудник факультета Вычислительной математики и кибернетики, Московский государственный университет (Москва, Российская Федерация), salnikov@cs.msu.ru.

*Поступила в редакцию 22 ноября 2013 г.*

***Bulletin of the South Ural State University***
***Series "Computational Mathematics and Software Engineering"***
***2014, vol. 3, no. 1, pp. 55–67***

# OPTIMIZATION OF MODELING IMMOBILIZED PROTEIN INTERACTIONS ON COMPULATIONAL CLUSTERS WITH THE SUPPLYING OF ACCESS TO THE ALGORITHM VIA WEB-INTERFACE

***K.V. Romanenkov,*** Lomonosov Moscow State University (Moscow, Russian Federation)***,***
***A.N. Salnikov,*** Lomonosov Moscow State University (Moscow, Russian Federation)

In this work were introduced parallel versions based on OpenMP and MPI technologies of sequential program for modeling immobilized protein interactions. Both versions have shown good scalability and better time indices to compare with the sequential version when running on single processor. Need to mention that modeling of immobilized protein interactions for some compounds have taken more than twenty hours of computations on several hundreds of processors, that's why for such modeling tasks with the great quantity of positions availability of nondeterministic algorithms, providing biologically correct result in reasonable time, seems to be rather important. Selection of stochastic algorithms has proved its value: both Monte-Carlo and simulated bee colony algorithms had found conformation corresponding minimal energy state. Supplying of access to the algorithm via web-interface measures up modern specifications of remote computations and allows the wide circle of specialists use computational power of Moscow State University and, taking into account extending the sphere of application tasks of molecular simulation, the presence of open web-interface providing remote access to the computational clusters is quite an important task.

*Keywords: bioinformatics, multiprocessor systems, immobilized protein interactions, parallel algorithms, stochastic algorithms*

## References


1. Fortenberry C., Bowman E.A., Proffitt W.,Dorr B., Combs S., Harp J., Mizoue L.,Meiler J. Exploring Symmetry as an Avenue to the Computational Design of Large Protein Domains // Journal of the American Chemical Society. 2011. Vol. 133, No. 45. P. 18026–18029.
2. Wernisch L., Hery S., Wodak S.J. Automatic protein design with all atom force-fields by exact and heuristic optimization // Journal of Molecular Biology. 2000. Vol. 301, No. 3. P. 713–736.
3. Leach A.R., Lemon A.P. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm // Proteins: Structure, Function, and Bioinformatics. 1998. Vol. 33, No. 2. P. 227–239.
4. Tantar A.A., Melab N., Talbi E.G., Parent B., Horvath D. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid // Future Generation Comp. Syst. 2009. Vol. 23, No. 3. P. 398–409.
5. Pitman D.J., Schenkelberg C.D., Huang Y.M., Teets F.D., DiTursi D., Bystroff C. Improving computational efficiency and tractability of protein design using a piecemeal approach. A strategy for parallel and distributed protein design. Bioinformatics. 2013. URL: `http://dx.doi.org/10.1093/bioinformatics/btt735` (accessed: 30.12.2013).
6. Molto G., Suarez M., Tortosa P., Alonso J.M., Hernandez V., Jaramillo A. Protein design based on parallel dimensional reduction // Journal of Chemical Information and Modeling. 2009. Vol. 49, No. 5. P. 1261-1271.
7. Street A.G., Mayo S.L Computational protein design. URL: `http://dx.doi.org/10.1016/S0969-2126(99)80062-8` (accessed: 24.02.2013).
8. Voigt C.A., Gordon D.B., Mayo S.L. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design // Journal of Molecular Biology. 2000. Vol. 299, No. 3. P. 789–803.
9. Pham D.T., Ghanbarzdeh A., Koc E., Otri S., Rahim S., Zaidi M. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems // 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS'2006). 2006. P. 454–461.
10. Hamers-Casterman C., Atarhouch T., Muyldermans S., Robinson G., Hamers C., Songa E.B., Bendahman N., Hamers R. Naturally occurring antibodies devoid of light chains // Nature. 1993. Vol. 363. P. 446–448.
11. Revets H.,De Baetselier P., Muyldermans S. Nanobodies as novel agents for cancer therapy // Expert Opinion on Biological Therapy. 2005. Vol. 5, No. 1. P. 111–124.
12. Grishin A., Fonfara I., Wende W., Alexeyevsky D., Alexeyevsky A., Spirin S., Zanegina O., Karyagina A. Bioinformatics analysis of LAGLIDADG homing endonucleases for construction of enzymes with changed DNA recognition specificity // 4-th Moscow Conference on Computational Molecular Biology. Moscow, MSU, 2009. P. 123.
13. Knjazev N.A., Salnikov A.N. Sistema spravedlivogo planirovanija i unificirovannogo zapuska zadach pol'zovatelja na superkomp'juterah [Fair scheduling system for running user's tasks on supercomputers in unified mode]. Parallel'nye vychislitel'nye tehnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferencii (Ufa, 29 marta – 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March, 29 – April, 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 665–666.

14. A.N. Salnikov. Internet-cervis dlja postroenija mnozhestvennogo vyravnivanija posledovatel'nostej na mnogoprocessornyh sistemah, sozdannyj na osnove data-flow modifikacii algoritma MUSCLE [Web-Service based on data-flow modification of MUSCLE algorithm for constructing multiple sequence alignment on multiprocessor systems]. Parallel'nye vychislitel'nye tehnologii (PaVT'2009): Trudy mezhdunarodnoj nauchnoj konferencii (Nizhni Novgorod, 30 marta – 3 aprelya 2009) [Parallel Computational Technologies (PCT'2009): Proceedings of the International Scientific Conference (Nizhni Novgorod, Russia, March, 30 – April, 3, 2009)]. Chelyabinsk, Publishing of the South Ural State University, 2009. P. 680–687.

*Received 22 November 2013*