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

К.В. Романенков, А.Н. Сальников

1. 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 гемоглобина содержится более 64000 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.

1.1. Принципы компьютерного моделирования

Основной целью компьютерного моделирования белковых соединений является выбор аминокислот в воспроизводимой структуре, минимизирующих общую энергию системы [3]. 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. энергия остова, остающаяся неизменной при поиске GMEC и поэтому не участвующая в оптимизации
2. энергия взаимодействия ротамер / остов
3. энергия взаимодействия ротамер / ротамер

$$\varepsilon(C) = E_{template} + \sum_{i=1}^{n} E(C_i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(C_i, C_j)$$

Форм. 1. Энергия конформации {C1, …, Cn| Ci — ротамер в i позиции}

1.2. Особенности задачи создания молекулярных интерфейсов

Необходимо отметить ключевые особенности данной области вычислительного моделирования белковых соединений:

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) [4]).
5. Проблема GMEC — NP-полная [4]

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.

1. DEE(Dead End Elimination) [4]. Простейший метод дискретной оптимизации, позволяет отсекать заведомо неперспективные ротамеры, основываясь на их вкладе в общую энергию.

2. MC/Q(Monte Carlo / Quenching). Несмотря на то, что метод был разработан в середине 50-х годов, успешно применяться в задачах моделирования белковых соединений он начал лишь в 90-х годах [4]. 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.

$$P = e^{\frac{-(E_{new} - E_{old})}{kT}}$$

Форм. 2. Вероятность Больцмана подстановки менее энергетически выгодного ротамера

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.

3. Алгоритм пчелиного поиска (Simulated Bee Colony) [7]. Алгоритм, появившийся в 2005 году, хорошо зарекомендовал себя как в задачах непрерывной, так и дискретной оптимизации. 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. Задачи были представлены институтом физико-химической биологии имени А. Н. Белозерского.

1. Белковый комплекс LAGLIDADG эндонуклеаз. 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.
2. Белковый комплекс антитело - антиген. 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:

   а) Оптимизация комплекса антитело — антиген средствами компьютерного моделирования для повышения аффинности антитела к антигену.

   б) Дизайн новых антител к указанным антигенам «de novo», имитирующий селективный отбор иммунной системы.

Относительно недавно были открыты и сейчас активно исследуются наноантитела[9, 10]: 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*6*191*308*190*106*196*326 флопов, то есть более 2,7 петафлопов.

2.3. Последовательная реализация

Авторам была предомставлена последовательная программа fitprot [5], написанная на языке python, которая осуществляла полный перебор конформаций для указанных пользователем позиций, предварительно применяя к ним фильтр DEE [см. обзор методов оптимизации, пункт 1]. На вход программе подаются два файла: со структурой соединения в формате PDB и с тестовым файлом, описывающим в каких позициях и у каких ротамеров надо моделировать мутацию остатка. Была произведена модификация программы fitprot так, чтобы её было возможно запускать на многопроцессорных системах как с общей памятью, так и кластерной архитектуры.

3. Реализация

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. В OpenMP реализации вся 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 [6]. 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 [8]. 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 рисунке 1.

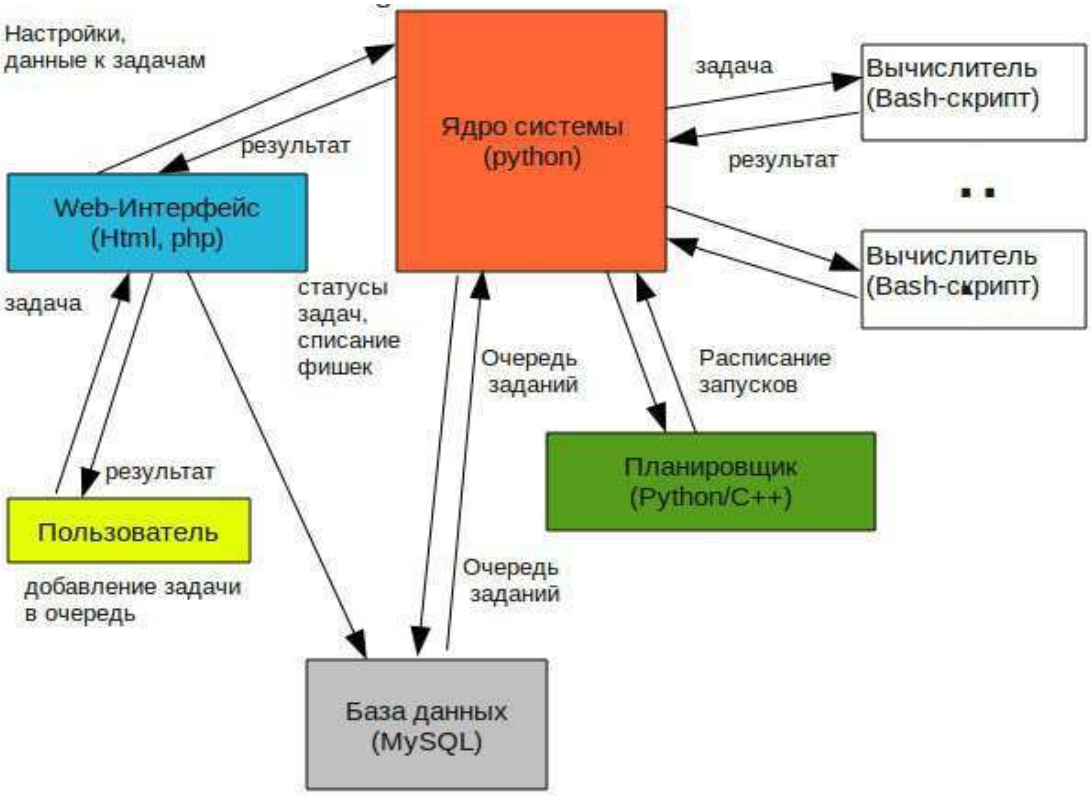

Рис. 1. Структура системы Aligner

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 рисунке 2.

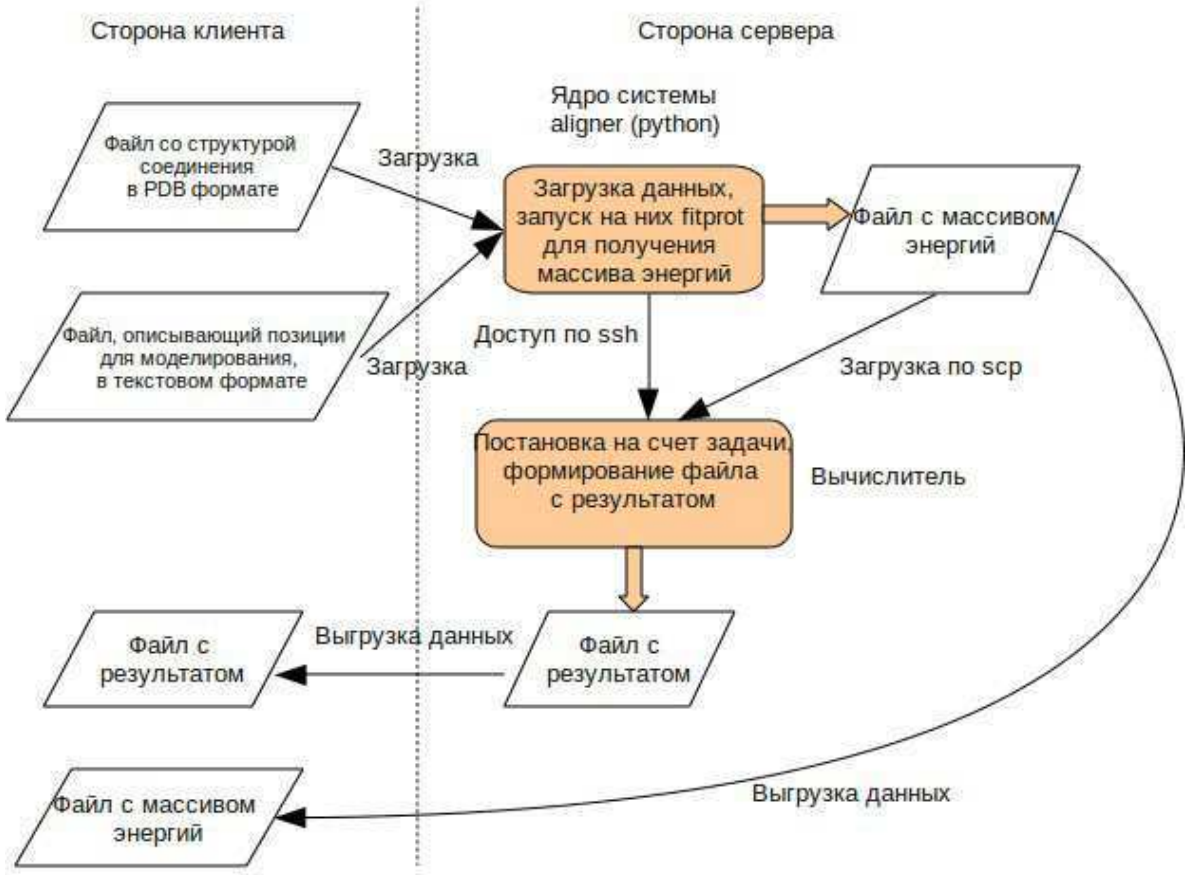

Рис. 2. Путь данных, при удаленном запуске заданий

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 гигабайтами общей памяти, и доступная из сети МГУ. Для расчетов молекулярного интерфейса наноантитела использовался комплекс «ЧЕБЫШЕВ», работающий на процессорах Intel Xeon E5472 3.0 GHz и имеющий пиковую производительность 60 терафлопов в секунду.

На рисунке 3, приведенном ниже, указаны ускорения параллельных версий детерминированного алгоритма.

Таблица 1. Времена работы(секунды) MPI версии программы для 3 позиций эндонуклеазы и OpenMP версии программы для 4 позиций эндонуклеазы

Процессоры	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

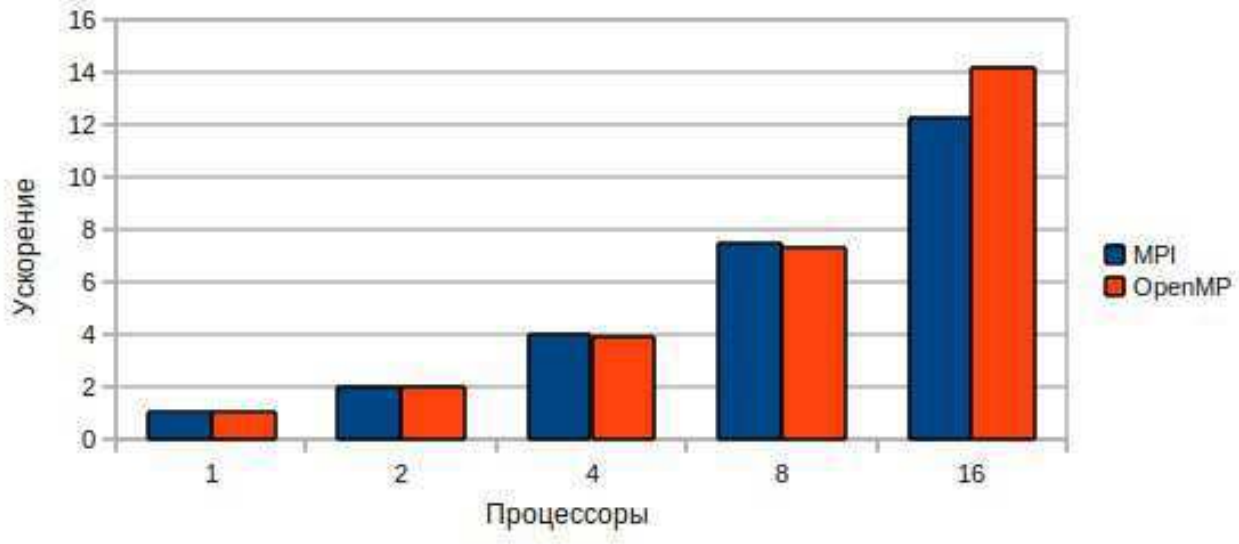

Рис. 3. Ускорение для параллельных версий детерминированного алгоритма

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 конформеры 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.

Таблица 2. Результаты работы алгоритма MC/Q

Структура	Нахождение GMEC	Количество найденных конформаций из первой десятки минимально возможных вариантов (не считая GMEC)
Эндонуклеазы, 3 позиции	да	3
Эндонуклеазы, 4 позиции	да	1
Эндонуклеазы, 4 позиции, увеличенное число ротамеров для каждой позиции	да	1
Наноантитело в комплексе с лизоцимом, 4 позиции	да	4
Наноантитело в комплексе с лизоцимом, 5 позиции	да	4

Таблица 3. Результаты работы алгоритма пчелиного поиска

Структура	Нахождение GMEC	Количество найденных конформаций из первой десятки минимально возможных вариантов (не считая GMEC)
Эндонуклеазы, 3 позиции (количество итераций уменьшено на порядок, чтобы предотвратить схождение всех решений к GMEC)	да	3
Эндонуклеазы, 4 позиции	да	4
Эндонуклеазы, 4 позиции, увеличенное число ротамеров для	да	3

каждой позиции		
Наноантитело в комплексе с лизоцимом, 4 позиции	да	3
Наноантитело в комплексе с лизоцимом, 5 позиции	да	4

6. Выводы

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 при запуске на 1 процессоре. 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.

Работа выполняется при поддержке грантов РФФИ: 11-07-00756-а, 11-07-00614-а и госконтракта по ФЦП "Научные и научно-педагогические кадры инновационной России" П1317.

ЛИТЕРАТУРА:

1. C. Fortenberry et al. (2011) *J. Amer. Chem. Soc.*, published online October 6, DOI: 10.1021/ja2051217
2. Wernisch L, Hery S, Wodak SJ. Automatic protein design with all atom force-fields by exact and heuristic optimization. J Mol Biol. 2000 Aug 18;301(3):713-36. PubMed PMID: 10966779
3. Street, A. G. & Mayo, S. L. (1999). Computational protein design. Structure, 7, R105-R109.
4. Voigt CA, Gordon DB, Mayo SL. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. J Mol Biol. 2000 Jun 9;299(3):789-803. PubMed PMID: 10835284.
5. Grishin A, Fonfara I, Wende W, Alexeyevsky D, Alexeevski 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, 2009, Moscow, MSU, p.123.
6. Н.А. Князев, А.Н. Сальников «Система справедливого планирования и унифицированного запуска задач пользователя на суперкомпьютерах» // Параллельные вычислительные технологии (ПаВТ’2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.) [Электронный ресурс] – Челябинск: Издательский центр ЮУрГУ, 2010. стр. 665–666
7. D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm — A Novel Tool for Complex Optimisation Problems, Proceedings of IPROMS 2006 Conference, pp. 454–461, 2006.
8. А.Н. Сальников «Интернет-сервис для построения множественного выравнивания последовательностей на многопроцессорных системах, созданный на основе data-flow модификации алгоритма MUSCLE» // Труды международной научной конференции Параллельные вычислительные технологии ПАВТ-2009, издательство ЮУрГУ г. челябинск стр. 680-687
9. Hamers-Casterman C, Atarhouch T, Muyldermans S, Robinson G, Hamers C, Songa EB, Bendahman N, Hamers R (1993) Naturally occurring antibodies devoid of light chains. Nature 363:446–448
10. Revets H, De Baetselier P, Muyldermans S (2005) Nanobodies as novel agents for cancer therapy. Expert Opin Biol Ther 5:111–124