

УДК 519.6

**TTDock: МЕТОД ДОКИНГА НА ОСНОВЕ ТЕНЗОРНЫХ ПОЕЗДОВ****Д. А. Желтков<sup>1</sup>, И. В. Офёркин<sup>2</sup>, Е. В. Каткова<sup>2</sup>, А. В. Сулимов<sup>2</sup>,  
В. Б. Сулимов<sup>3</sup>, Е. Е. Тыртышников<sup>4</sup>**

Разработан метод докинга на основе тензорных поездов, позволяющий с высокой вероятностью найти положение, в котором достигается глобальный минимум энергии системы белок–лиганд. Полученный метод сравнивается с одним из наиболее эффективных методов докинга на основе генетических алгоритмов SOL. Результаты тестирования показывают, что метод на основе тензорных поездов в 10 раз менее затратен при сопоставимой надежности получения глобального оптимума энергии.

**Ключевые слова:** тензорный поезд, крестовый метод, глобальная оптимизация, докинг, компьютерный дизайн лекарств.

**1. Введение.** Задача расчета энергии связывания белка и лиганда (органической молекулы — кандидата в ингибиторы этого белка-мишени) является одной из ключевых задач применения компьютерного молекулярного моделирования для разработки новых лекарственных средств [1]. Эффективным решением этой задачи представляется процедура докинга, при которой на многомерной энергетической поверхности системы белок–лиганд находится то положение лиганда, при котором энергия этой системы минимальна, т.е. докинг — это нахождение глобального минимума энергии системы белок–лиганд. Правильное позиционирование лиганда в белке-мишени дает также и оценку энергии связывания лиганда с белком.

Докинг является сравнительно быстрым методом и во многих случаях хорошо справляется с задачей позиционирования лиганда в белке, однако он зачастую не способен с адекватной точностью предсказывать величину свободной энергии связывания. Причин тому много. Прежде всего отметим неадекватное описание межмолекулярных взаимодействий с помощью существующих классических силовых полей (классических потенциалов), описывающих взаимодействия между атомами как взаимодействия между классическими частицами; как хорошо известно, такие взаимодействия могут быть адекватно описаны только с помощью методов квантовой механики. Кроме того, отметим и большое число приближений, которые делаются для того, чтобы задачу докинга сделать решаемой — ведь даже для сравнительно небольшого ингибитора, имеющего, скажем, только 10 внутренних вращательных степеней свободы, глобальный минимум надо искать на 16-мерной энергетической поверхности, даже если считать белок-мишень жестким, а это, очевидно, очень непростая задача, которую невозможно решить методом простого перебора.

Один из таких существенных и широко распространенных подходов, позволяющих существенно ускорить процесс докинга, состоит во введении заранее рассчитываемой сетки потенциалов, описывающих взаимодействие лиганда с белком [2, 3]. Этот подход, однако, не позволяет проводить локальную оптимизацию в процессе докинга, учитывать при докинге подвижность атомов белка и адекватно описывать нелокальные взаимодействия белка, лиганда и их комплекса с водой.

Кроме того, в большинстве программ докинга выводы о величине энергии связывания лиганда с белком-мишенью и соответствующей ей экспериментально наблюдаемой константе ингибирования делаются на основе найденного значения потенциальной энергии системы белок–лиганд в глобальном минимуме. Однако такая статическая картина не соответствует действительности — константа ингибирования определяется не потенциальной энергией, а минимумом свободной энергии Гиббса системы белок–лиганд,

<sup>1</sup> Московский государственный университет им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, д. 1, стр. 52, 119992, Москва; аспирант, e-mail: dmitry.zheltkov@gmail.com

<sup>2</sup> ООО “Димонта”, ул. Нагорная, д. 15, корп. 8, 117186, Москва; И. В. Офёркин, программист, e-mail: io@dimonta.com; Е. В. Каткова, мл. науч. сотр., e-mail: katkova@dimonta.com; А. В. Сулимов, системный программист, e-mail: sulimovv@mail.ru

<sup>3</sup> Научно-исследовательский вычислительный центр, Московский государственный университет им. М.В. Ломоносова, Ленинские горы, д. 1, стр. 4, 119992, Москва; зав. лабораторией, e-mail: vladimir.sulimov@gmail.com

<sup>4</sup> Институт вычислительной математики РАН, ул. Губкина, д. 8, 119333, Москва; директор, e-mail: eugene.tyrtysnikov@gmail.com

имеющей как энтальпийную, так и энтропийную составляющие, первая из которых в основном определяется именно потенциальной энергией, а вторая имеет существенно статистическую природу. Более точно константа ингибирования определяется разностью свободной энергии Гиббса системы в двух состояниях: в связанном, когда образован комплекс белок–лиганд, и в свободном, когда белок и лиганд друг с другом не взаимодействуют, причем энтальпийная и энтропийная составляющие этой разности нередко близки по величине (или имеют один порядок), но часто противоположны по знаку.

Наличие нескольких способов связывания лиганда с белком, т.е. существование нескольких минимумов близких по потенциальной энергии к глобальному минимуму, непосредственно влияет на энтропию комплекса лиганд–белок: энтропия растет пропорционально логарифму количества локальных минимумов с низкой потенциальной энергией связывания. Поэтому константа связывания лиганда с белком зависит не только от положения лиганда с наименьшей потенциальной энергией, но и от всех других возможных положений связывания лиганда с достаточно низкой энергией (в пределах нескольких  $kT$  от энергии глобального минимума). Таким образом, для того чтобы правильно оценить энергию связывания, необходимо иметь представление не только об одном отдельно взятом положении лиганда в белке, соответствующем глобальному минимуму потенциальной энергии системы белок–лиганд, но и о форме и глубине всего конформационного фазового пространства этой системы [4].

Таким образом, задача докинга, целью которой является расчет свободной энергии связывания лиганда с белком и определяемой с ее помощью константы ингибирования, представляет собой нахождение не только глобального минимума потенциальной энергии системы белок–лиганд, но и всех находящихся рядом с ним по энергии (в пределах нескольких  $kT$ ) локальных минимумов, в которых рассматриваемая система белок–лиганд также может находиться благодаря тепловым флуктуациям.

Для более аккуратного расчета свободной энергии связывания лиганда с белком-мишенью необходимо отказаться от приближения, в котором заранее рассчитывается сетка потенциалов пробных атомов лиганда с белком, а также находить не только глобальный минимум потенциальной энергии системы белок–лиганд, но и все близкие к нему по энергии локальные минимумы. Все это, в конечном счете, требует существенного ускорения алгоритма докинга по сравнению с используемыми в настоящее время алгоритмами.

В настоящей статье предложен новый алгоритм нахождения глобального минимума, основанный на использовании новейших методов работы с тензорами большой размерности, которые должны существенно ускорить решение задачи докинга. Большая размерность тензоров при докинге естественно возникает следующим образом. Для достижения хорошей точности позиционирования лиганда в белке по каждой из, например, 20 степеней свободы лиганда необходимо, по крайней мере, 100 точек. Таким образом, число точек, среди которых надо найти ту, в которой энергия системы самая низкая, равно  $(10^2)^{20} = 10^{40}$ .

В рассматриваемой работе разработан и реализован в виде программы TTDock новый алгоритм докинга, основанный на использовании разложения тензоров большой размерности в формате тензорного поезда (далее — ТТ-разложения). Проведено его сравнение с методом, реализованным в программе докинга SOL на основе генетического алгоритма, являющегося одним из лучших в данной области. Показано, что методу докинга на основе ТТ-разложений для нахождения глобального оптимума необходимо всего несколько миллионов или несколько десятков миллионов вычислений функции для вычисления глобального оптимума с вероятностью 0.9, в то время как методу программы докинга SOL необходимо в 10 раз больше вычислений целевой функции для получения результата с той же надежностью.

## 2. Докинг на основе ТТ-разложения.

**2.1. ТТ-разложение.** В многомерных массивах (тензорах) число элементов растет экспоненциально с ростом числа измерений. Например, для хранения всех элементов тензора  $A \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$  размерности  $d = 10$  при одном и том же числе отсчетов по каждому измерению  $n_i = 100$  требуется  $100^{10} = 10^{20}$  ячеек памяти, что крайне много. В этой связи важно иметь компактные представления таких объектов.

Для тензоров наиболее известны два малопараметрических формата: каноническое разложение и разложение Таккера [5]. Для хранения тензора  $A \in \mathbb{R}^{n_1 \times \dots \times n_d}$  в каноническом формате требуется  $\left(\sum_{i=1}^d n_i\right)R$  ячеек памяти, где  $R$  — так называемый канонический ранг данного разложения. Однако надежных алгоритмов аппроксимации тензора в каноническом представлении не существует. Хранение тензора в формате Таккера требует  $\sum_{i=1}^d n_i r_i + \prod_{i=1}^d r_i$  ячеек памяти, где  $r_i$  — ранги разложения Таккера. Для разложения Таккера надежные алгоритмы приближения тензора в данном представлении существуют. Более того, это можно сделать с помощью крестового алгоритма интерполяции [6], который позволяет найти приближе-

ние, вычислив лишь относительно небольшое число элементов тензора. Тем не менее, разложение Таккера нельзя применять при больших  $d$ , так как количество ячеек для его хранения растет экспоненциально по  $d$  (“проклятие размерности”).

В 2009 г. в вычислительную математику был введен новый формат представления тензоров — тензорный поезд (Tensor Train, ТТ) [7]. Этот формат представляет собой разложение, в котором исходный тензор размерности  $d$  сводится к  $d$  тензорам размерности 3:

$$A(i_1; i_2; \dots; i_d) = \sum_{\alpha_0=1; \dots; \alpha_d=1}^{r_0; \dots; r_d} G_1(\alpha_0; i_1; \alpha_1) \dots G_d(\alpha_{d-1}; i_{d-1}; \alpha_d).$$

Тензоры  $G_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$  называются вагонами (иногда ядрами) тензорного поезда, а числа  $r_k$  — его рангами (ТТ-рангами). При этом по определению  $r_0 = r_d = 1$ . Для хранения тензора в ТТ-формате требуется всего  $\sum_{i=1}^d n_i r_{i-1} r_i$  ячеек памяти.

Важно то, что в ТТ-формате существуют быстрые алгоритмы для основных операций над тензорами, большинство операций выполняется за  $O(dnr^3)$  арифметических действий, где  $r = \max_{i \in [1, d-1]}(r_i)$ ,  $n = \max_{i \in [1, d]}(n_i)$ . Есть быстрый и надежный метод ТТ-округления [8], позволяющий получать приближения тензора с малыми ТТ-рангами на основе SVD-алгоритма (Singular Value Decomposition), применяемого для вычисления сингулярных разложений матриц развертки, ассоциированных с ТТ-разложением этого тензора. Матрицы развертки в явном виде не строятся, а компоненты сингулярных разложений также получаются в специальных параметризациях, определенных структурой ТТ-разложения. Другие итерационные алгоритмы аппроксимации сводят задачу к последовательности локальных задач аппроксимации в духе метода переменных направлений (ALS — Alternating Least Squares), в их числе возникший в теоретической физике метод DMRG (Density Matrix Renormalization Group). Изучение ТТ-разложения с позиций современной вычислительной линейной алгебры привело к переосмыслению алгоритмов, применявшихся в квантовой физике, и к появлению совершенно новых алгоритмов. Среди них для нас особенно важен метод ТТ-CROSS [9], получающий приближение тензора в ТТ-формате весьма экономичным образом: на вычисление  $O(dnr^2)$  его элементов затрачивается  $O(dnr^3)$  операций. Метод ТТ-CROSS является обобщением крестового метода интерполяции матрицы по ее столбцам и строкам [10–12].

Часто тензор задается как функция от целочисленных аргументов, легко вычисляется одно или несколько значений этой функции, но в силу “проклятия размерности” невозможно вычислить ее значения во всех точках  $d$ -мерной сетки. Алгоритмы ТТ-крестовой аппроксимации позволяют получать структурированные малопараметрические представления матриц и тензоров, заданных в виде функций от целочисленных переменных. С их помощью можно решать множество задач: многомерное интегрирование, решение интегральных уравнений, задачи спиновой динамики, квантовой химии и др. Мы используем конструкции ТТ-CROSS для построения нового метода оптимизации функционала от  $d$  переменных, в котором число вычислений значений функционала зависит от  $d$  линейно. Естественно, применение этого метода основано на существенных априорных предположениях о свойствах целевого функционала. В данной работе метод применяется к задаче докинга. Теоретическое исследование общих предположений о свойствах целевого функционала является предметом дальнейших исследований.

Замечено, что матричный крестовый метод получает приближение к максимальному по модулю элементу матрицы [13]. Тем же свойством обладает и метод ТТ-CROSS, причем приближение к максимальному по модулю элементу тензора находится весьма быстро — за  $O(dnr^3)$  арифметических операций. Данное наблюдение является отправной точкой для разработки метода быстрого поиска глобального оптимума для функций, имеющих малый ТТ-ранг.

Исходная задача оптимизации заменяется на эквивалентную задачу поиска максимума по модулю. Для этого исходная функция непрерывно и монотонно отображается на часть интервала  $[0; +\infty)$ . Для построения этого отображения можно использовать, например, функции  $\exp(-x)$  или  $\text{arcctg}(x)$ , если исходная задача является задачей минимизации, и  $\exp(x)$  или  $\text{arcctg}(-x)$  в задаче максимизации. Произведя дискретизацию, получаем модифицированный тензор. С помощью ТТ-крестового метода находим его малоранговое приближение в ТТ-формате и одновременно выполняем поиск максимального по модулю элемента тензора. Поскольку значения отображения неотрицательны, максимальный по модулю элемент будет максимальным элементом на введенной сетке. Таким образом, если сетка была выбрана достаточно мелкой, то решение задачи глобальной оптимизации будет найдено. Отметим, что сложность такого метода составляет  $O(dnr^3)$  арифметических операций и  $O(dnr^2)$  операций вычисления функции.

Очевидно, что метод целесообразно применять для относительно малых значений  $r$ . В то же время достаточно точное приближение исходного тензора может приводить к большим ТТ-рангам. Тем не менее метод ТТ-CROSS в ходе своей работы осуществляет поиск больших по модулю элементов. В настоящей статье предложен разработанный на его основе метод глобальной оптимизации, который строит лишь очень грубое приближение тензора и с его помощью находит приближение к глобальному максимуму. Отметим, что метод, получающийся таким образом, имеет некоторую связь с генетическими алгоритмами. Вычислительные затраты составляют  $O(dnr^3)$  арифметических операций,  $O(dnr^2)$  вычислений функции и  $O(dr)$  локальных оптимизаций. Однако “хорошего” приближения ТТ-ранга  $r$  может и не существовать. В нашем методе  $r$  — это некоторый автоматически подбираемый параметр, который обычно много меньше ТТ-ранга “достаточно хорошего” приближения.

Так как основная задача докинга (поиск минимумов вблизи глобального) является задачей глобальной оптимизации, то изложенные методы на основе ТТ могут рассматриваться как альтернатива существующим методам докинга.

Методы оптимизации, основанные на ТТ, активно используют структуру оптимизируемой функции. В случаях, когда оптимизируемые функции имеют хорошую, с точки зрения ТТ, структуру, поиск оптимума может осуществляться заметно быстрее и/или надежнее, чем решение этой задачи с помощью других известных методов.

**2.2. Целевая функция: вычисление энергии лиганда в поле белка.** В качестве реперного алгоритма докинга в данной работе используется генетический алгоритм (GA), являющийся одним из самых эффективных алгоритмов докинга гибких лигандов и реализованный в таких популярных программах докинга, как AutoDock [14], GOLD [15], MolDock [16], а также в программе докинга SOL [2, 3]. Для определенности в данной работе белок-мишень считается жестким, а при поиске глобального минимума в качестве целевой функции используется энергия лиганда, вычисляемая в рамках силового поля MMFF94 с помощью сетки заранее рассчитанных с помощью программы SOLGRID [2, 3] потенциалов взаимодействия белка-мишени с пробными атомами лиганда. При этом появляется возможность сравнить и эффективность, и быстродействие программы докинга TTDock с соответствующими характеристиками программы докинга SOL, поскольку они имеют одинаковые целевые функции. Остановимся подробнее на приближениях, используемых в программах SOLGRID и SOL.

1. Белок предполагается жестким, а для имитации подвижности атомов белка вводится уширение потенциалов [2], составляющее в данной работе  $0.4 \text{ \AA}$ . Докинг лигандов при этом осуществляется в куб докинга (он имеет длину ребра  $22 \text{ \AA}$ ), охватывающий (с большим запасом) активный центр белка-мишени. В этом кубе в узлах трехмерной сетки, содержащей 101 точку по каждому измерению, записаны потенциалы взаимодействия атомов лиганда с атомами всего белка, включая те атомы белка, которые находятся вне куба докинга. В каждом узле записаны значения различных потенциалов взаимодействия каждого типа пробных атомов лиганда: электростатическое взаимодействие, вандерваальсово взаимодействие, потенциалы десольватации, а типы пробных атомов лиганда относятся к типам, определяемым силовым полем MMFF94 [17].

2. Используется упрощенная типизация атомов: некоторые типы атомов, являющиеся различными в соответствии с типизацией силового поля MMFF94, объединяются в единые типы на основе некоторых интуитивных соображений, а вместо 99 возможных сеток потенциалов для вандерваальсовых взаимодействий создается всего 27 сеток потенциалов для 27 различных упрощенных типов пробных атомов [2].

3. При вычислении энергии десольватации влияние растворителя учитывается при помощи упрощенной обобщенной модели Борна [18], в которой упрощения вводятся, чтобы нелокальную задачу электростатического экранирования свести к локальным потенциалам на сетке.

4. При позиционировании лиганда в процессе работы программ докинга SOL и TTDock локальная оптимизация энергии по положениям атомов лиганда не проводится.

5. Длины связей и валентные углы между связями лиганда в процессе докинга с помощью программ SOL и TTDock остаются неизменными; лиганд может изменять свое положение при помощи торсионных вращений вокруг одинарных связей, а также за счет вращения и трансляции его как целого.

6. Атомы белка создают статичное поле потенциалов взаимодействия с атомами лиганда (электростатическое, вандерваальсово, поле потенциалов эффектов десольватации). Когда лиганд занимает некоторое фиксированное положение относительно белка, атомы лиганда приобретают в каждом из этих полей определенную энергию, а весь лиганд приобретает энергию в виде суммы энергий составляющих его атомов и энергии взаимодействия атомов лиганда между собой.

Реализованный в программе докинга SOL генетический алгоритм (GA) [2, 3] является математическим методом нахождения глобального минимума, в основе которого лежит процесс моделирования эво-

люции некой популяции особей по Дарвину с учетом генетических механизмов (мутаций, кроссинговера и др.). Особи — это положения лиганда в активном центре белка-мишени. Критерием выживаемости особи является энергия лиганда в поле потенциалов белка — чем ниже эта энергия, тем вероятнее выживает эта особь.

Так как GA — это эвристический алгоритм и он не может гарантировать нахождение глобального минимума энергии, то для контроля достоверности полученных результатов проводится несколько независимых запусков GA. Результаты каждого запуска запоминаются и после проведения всех независимых запусков GA определяется, насколько похожи их результаты между собой. При достаточно близких результатах, полученных в разных независимых запусках GA, они считаются достоверными. Тем не менее, нередко множественные запуски GA приводят к нескольким отличающимся как по геометрии, так и по энергии положениям лиганда. В таком случае можно предположить, что помимо глобального минимума (т.е. результата с минимальной энергией) программа находит также несколько близких локальных минимумов, учет которых также дает вклад в свободную энергию взаимодействия белок–лиганд.

**2.3. Алгоритм докинга на основе тензорных поездов.** Алгоритм докинга на основе тензорных поездов базируется на матричном крестовом методе [10–12] и его многомерном обобщении — методе TT-CROSS [9]. Матричный крестовый метод позволяет получить приближение матрицы  $A \in \mathbb{R}^{m \times n}$  с относительной точностью  $\varepsilon$ , вычислив всего  $O((m+n)r)$  элементов матрицы и выполнив  $O((m+n)r^2)$  операций, где  $r$  — ранг приближения, определяемый в ходе работы метода. Отметим, что ранг  $r$ , как правило, близок к минимальному рангу, необходимому для приближения матрицы с заданной точностью  $\varepsilon$ , и существуют теоремы о квазиоптимальности этого метода [10–12]. Рассматриваемый метод показал свою эффективность при решении интегральных уравнений [11] и многих других задач.

Метод TT-CROSS находит приближение тензора  $T \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , используя всего  $O(dnr^2)$  элементов тензора и выполняя  $O(dnr^3)$  операций, где  $n = \max_{1 \leq i \leq d} n_i$ ,  $r = \max_{0 \leq i \leq d} r_i$ ,  $r_i$  — TT-ранги полученного приближения. Как и в матричном случае, ранги определяются в ходе работы метода. Метод TT-CROSS применяется при многомерном интегрировании, численном решении многомерных интегральных уравнений, решении задач квантовой химии и др.

Отметим, что оба метода в ходе своей работы осуществляют поиск больших по модулю элементов матрицы или тензора соответственно. Это позволяет использовать их в качестве основы для методов оптимизации в целом и для методов решения задачи докинга в частности. Заменяем основную задачу докинга (поиск локальных минимумов, включая глобальный) на эквивалентную задачу поиска локальных максимумов по модулю. Для этого отобразим исходную функцию  $e(\bar{v})$  непрерывным монотонно убывающим отображением  $f(x)$ , имеющим малый тензорный ранг, на подобласть области  $[0; +\infty)$ . В качестве функционалов отображения можно использовать, например, функции  $f(x) = \exp(-x)$  или  $f(x) = \text{arctg}(x)$ . После этого выполним интерполяцию отображенной функции  $g(\bar{v}) = f(e(\bar{v}))$  с помощью метода TT-CROSS с TT-рангами не выше некоторого  $r_{\max}$ . В результате работы алгоритма получим набор узлов интерполяции. Оптимизируя локально эти точки, находим набор локальных минимумов функции  $e(\bar{v})$ , содержащих с большой вероятностью глобальный, как это показывают эксперименты.

Отметим, что выбор функции отображения нетривиален. Например, при использовании в качестве отображения функции  $f(x) = \text{arctg}(x)$  максимумы итоговой оптимизируемой функции  $g(\bar{v})$  становятся едва различимыми, если минимумы исходной функции  $e(\bar{v})$  достаточно велики по модулю. При использовании на практике функции  $f(x) = \exp(-x)$  большие по модулю значения исходной функции преобразуются в машинные нули или бесконечности, что затрудняет функционирование метода TT-CROSS, а при большом их количестве делает его использование невозможным. Таким образом, на практике используются комбинации функций. Кроме того, естественной идеей является сдвиг функции отображения на текущее приближение к оптимуму для устранения указанных недостатков этих функций. Иными словами, оптимизируемый функционал должен выглядеть следующим образом:  $g_\alpha(\bar{v}) = f(e(\bar{v}) - \alpha)$ , где  $\alpha$  — найденное на данный момент приближение к глобальному минимуму функции  $e(\bar{v})$ .

Для упрощения изложения будем считать, что область, в которой осуществляется оптимизация, представляет собой  $d$ -мерный единичный куб  $[0, 1]^d$ . Введем сетку с  $n$  узлами по каждому из  $d$  направлений. Тензор значений функции  $g_\alpha$  на этой сетке обозначим через  $A^{(\alpha)}$ . Матрицами разверток данного тензора являются матрицы  $A_k^{(\alpha)} \in \mathbb{R}^{n^k \times n^{d-k}}$  и  $A_k^{(\alpha)}(i_1 i_2 \dots i_k, i_{k+1} i_{k+2} \dots i_d) = A^{(\alpha)}(i_1, i_2, \dots, i_d)$ . Алгоритм основан на последовательном приближении подматриц этих матриц разверток крестовым методом и локальной оптимизации полученных данным методом узлов интерполяции.

При переходе к рассмотрению матрицы  $A_k^\alpha$  от матрицы  $A_{k-1}^\alpha$  (шаг слева направо) имеются полученные на предыдущем шаге номера строк  $I_{k-1}$  и столбцов  $J_{k-1}$  матрицы  $A_{k-1}^\alpha$ . Каждый номер строки  $i \in I_{k-1}$  соответствует набору индексов  $(i_1, i_2, \dots, i_{k-1})$  исходного тензора  $A^\alpha$ ,  $i = i_1 + i_2 n + \dots + i_{k-1} n^{k-2}$ .

Аналогично, каждый номер столбца  $j \in J_{k-1}$  соответствует набору индексов  $(i_k, i_{k+1}, \dots, i_d)$  исходного тензора  $A^\alpha$ ,  $j = i_k + i_{k+1}n + \dots + i_d n^{d-k-1}$ .

Опишем получение наборов номеров строк  $I_k$  из  $I_{k-1}$ . Из  $I_k$  удаляются все имевшиеся элементы, а для каждого номера строки  $i \in I_{k-1}$  матрицы  $A_{k-1}^\alpha$ , соответствующего набору индексов  $(i_1, i_2, \dots, i_{k-1})$  тензора  $A^\alpha$ ,  $i = i_1 + i_2 n + \dots + i_{k-1} n^{k-2}$ , в  $I_k$  добавляется набор номеров строк  $i^{(1)}, i^{(2)}, \dots, i^{(n)}$ , соответствующих наборам индексов  $(i_1, i_2, \dots, i_{k-1}, 1), (i_1, i_2, \dots, i_{k-1}, 2), \dots, (i_1, i_2, \dots, i_{k-1}, n)$  тензора  $A^\alpha$ ,  $i^{(l)} = i_1 + i_2 n + \dots + i_{k-1} n^{k-2} + l n^{k-1}$ ,  $l = 0, 1, \dots, n - 1$ .

Построение столбцов  $J_k$  осуществляется иначе. Если мощность множества наборов строк  $I_k$  больше или равна максимальному возможному числу столбцов матрицы  $A_k^\alpha$ , т.е.  $|I_k| \geq n^{d-k}$ , то в  $J_k$  записываются все возможные номера строк,  $J_k = \{0, 1, \dots, n^{d-k} - 1\}$ . В противном случае в  $J_k$  все имевшиеся элементы сохраняются, а для каждого столбца  $j \in J_{k-1}$ , соответствующего набору индексов  $(i_k, i_{k+1}, \dots, i_d)$  тензора  $A^\alpha$ ,  $j = i_k + i_{k+1}n + \dots + i_d n^{d-k-1}$ , к ним добавляется индекс  $\tilde{j}$ , соответствующий набору индексов  $(i_{k+1}, \dots, i_d)$  тензора  $A^\alpha$ ,  $\tilde{j} = i_{k+1} + \dots + i_d n^{d-k-2}$ . Далее, пока в наборе  $J_k$  элементов меньше, чем в наборе  $I_k$ , он дополняется случайными номерами столбцов.

При переходе от матрицы  $A_{k+1}^\alpha$  к матрице  $A_k^\alpha$  (шаг справа налево) получение набора номеров столбцов  $J_k$  на основе набора номеров строк  $J_{k+1}$  выполняется аналогично получению наборов номеров строк  $I_k$  при шагах слева направо, а получение набора номеров строк  $I_k$  на основе набора номеров строк  $I_{k+1}$  выполняется аналогично получению наборов номеров столбцов  $J_k$  при шагах слева направо.

Добирать наборы  $J_k$  при шагах слева направо (наборы  $I_k$  при шагах справа налево) случайными столбцами (строками) можно не до мощности множества  $I_k$  ( $J_k$ ), но и до других разумных количеств. Например, можно добирать столбцы (строки) до числа  $2nr_{\max}$ . Это незначительно влияет на сложность алгоритма (и не влияет на асимптотическую сложность), однако для матрицы развертки  $A_1^\alpha$  ( $A_{d-1}^\alpha$ ) рассматривается существенно больше столбцов (строк), что приводит к более быстрой сходимости и большей надежности метода. Особенно это полезно, если число узлов сетки  $n$  по каждому направлению мало. Тестировался, однако, алгоритм с описанной ранее методикой дбора строк (столбцов).

В матрице  $A_k^\alpha$  рассматривается подматрица  $A(I_k, J_k)$ , где  $I_k$  и  $J_k$  сформированы описанным выше образом. Эта матрица приближается с помощью матричного крестового метода. Отметим, что в предлагаемом методе оптимизации полученное приближение не сохраняется; таким образом, можно считать, что крестовый метод лишь находит позиции узлов интерполяции. Наборы номеров строк  $I_k$  и столбцов  $J_k$  перезаписываются строковыми и столбцовыми значениями этих позиций соответственно.

Алгоритм 1

Оптимизация на основе тензорных поездов, инициализация

**Require:** размерность задачи  $d \in \mathbb{N}$ ,  $d \geq 2$ , число узлов сетки в каждом направлении  $n \in \mathbb{N}$ ,  $n \geq 2$ , оптимизируемая функция  $e \in \mathbb{R}^d \rightarrow \mathbb{R}$ .

**Ensure:**  $M$  — множество точек локальных минимумов, включая глобальный.

**procedure** TT\_Opt ( $d, n, e$ )

<b>for</b> $l \leftarrow 0, d$ <b>do</b>	Инициализация
$I_l \leftarrow \emptyset$	Множества столбцов и строк, содержащих узлы
$J_l \leftarrow \emptyset$	интерполяции, изначально пусты.
<b>end for</b>	
$I_0 \leftarrow J_d \leftarrow \{0\}$	
$\alpha \leftarrow 0$	Изначально сдвиг функционала равен 0.
$M \leftarrow \emptyset$	Множество найденных локальных минимумов пусто.

Алгоритм оптимизации на основе тензорных поездов, разбитый на три логических части — инициализация, шаги слева направо и шаги справа налево, представлен в алгоритмах: 1 (инициализация), 2 (шаги справа налево) и 3 (шаги слева направо). Действия между проверками критерия останова называем одной итерацией алгоритма. Сложность одной итерации составляет  $O(dnr_{\max}^3)$  операций,  $O(dnr_{\max}^2)$  вычислений функции и  $O(dr_{\max})$  локальных оптимизаций. Алгоритм в ходе работы использует  $O(nr^2)$  ячеек памяти.

Подобрать ранг  $r_{\max}$  и число итераций  $N$  можно следующим образом. Пусть имеются некоторые значения  $r$  и  $N$  этих величин. Если метод за  $N$  итераций при ограничении на ранг  $r_{\max}$  находит лучший глобальный минимум меньше, чем за  $N/2$  итераций, то увеличиваем  $N$  вдвое. Иначе увеличиваем  $r_{\max}$  вдвое. Продолжаем действовать таким образом, пока найденный минимум с ограничением ранга  $r_{\max}$

будет меньше, чем минимум, найденный с ограничением ранга  $r_{\max}/2$ .

Отметим, что алгоритм построен на основе метода TT-CROSS и во многом с ним совпадает. Если сохранять получаемые им данные, то можно получать некоторые (возможно, довольно грубые) приближения тензора, выполнив небольшое число дополнительных вычислений.

Алгоритм 2

Оптимизация на основе тензорных поездов, шаги справа налево

```

for  $k \leftarrow d - 1, 1$  do
   $g_\alpha(\bar{v}) \equiv f(e(\bar{v}) - \alpha)$ 
   $J_k \leftarrow \emptyset$ 
  for  $i \leftarrow 0, n - 1$  do
    for  $j \leftarrow 0, |J_{k+1}| - 1$  do
       $J_k \leftarrow J_k \cup \{J_{k+1}[j]n + i\}$ 
    end for
  end for
  if ( $n^k \leq |J_k|$ )
     $I_k \leftarrow \{0, 1, \dots, n^k - 1\}$ 
  else
    for  $j \leftarrow 0, |I_{k+1}| - 1$  do
       $I_k \leftarrow I_k \cup \{I_{k+1}[j] \bmod n^k\}$ 
    end for
    while  $|I_k| < |J_k|$ 
       $I_k \leftarrow I_k \cup \{\text{rand}(n^k - 1)\}$ 
    end while
  end if
   $(I_k, J_k) \leftarrow \text{CROSS}(A_k^\alpha(I_k, J_k), |I_k|, |J_k|, r_{\max}, \varepsilon)$ 
  for  $l \leftarrow 0, |I_k| - 1$  do
     $v_l \leftarrow \text{TO\_POINT}(d, n, k, I_k[l], J_k[l])$ 
     $v_l \leftarrow \text{OPTIMIZE}(d, e, v_l)$ 
     $M \leftarrow M \cup \{v_l\}$ 
  end for
  for  $l \leftarrow 0, |I_k| - 1$  do
     $(i, j) \leftarrow \text{TO\_GRID}(d, n, k, v_l)$ 
     $I_k \leftarrow I_k \cup \{i\}$ 
     $J_k \leftarrow J_k \cup \{k\}$ 
  end for
   $\alpha = \min_{m \in M} e(m)$ 
end for
until критерий остановки
return  $M$ 
end procedure

```

На основе изложенного метода оптимизации легко построить метод докинга. Для этого введем для каждой из  $d$  степеней свободы лиганда равномерную сетку с  $n$  узлами. Тензор значений энергии лиганда в поле белка на этой сетке содержит  $n^d$  элементов. Применяв метод оптимизации на основе тензорных поездов к этому тензору, получим положение лиганда, в котором достигается глобальный минимум энергии. Полученный метод докинга на основе тензорных поездов реализован на языке C++ в программе докинга TTDock. В качестве метода локальной оптимизации используется реализация симплекс-метода библиотеки GSL (GNU Scientific Library, <http://www.gnu.org/software/gsl>)."

**3. Сравнение результатов докинга программами SOL и TTDock.** Для сравнения программ SOL и TTDock необходимо выбрать критерии сравнения, которыми, очевидно, должны быть эффективность и быстрдействие. Казалось бы, что для докинга эффективность — это способность находить экспериментальное положение лиганда в активном центре белка, которое в настоящее время в основном определяется методами рентгеноструктурного анализа закристаллизованного комплекса белок–лиганд. Соответствующие данные помещаются в базу данных Protein Data Bank [19], содержащую к настоящему

## Алгоритм 3

Оптимизация на основе тензорных поездов, шаги слева направо

<pre> repeat   for <math>k \leftarrow 1, d-1</math> do     <math>g_\alpha(\bar{v}) \equiv f(e(\bar{v}) - \alpha)</math>     <math>A^\alpha(i_1, \dots, i_d) \equiv g_\alpha\left(\frac{i_1-1}{n-1}, \dots, \frac{i_d-1}{n-1}\right)</math>     <math>A_k^\alpha(i_1 \dots i_k, i_{k+1} \dots i_d) \equiv A_\alpha(i_1, \dots, i_d)</math>     <math>I_k \leftarrow \emptyset</math>     for <math>i \leftarrow 0, n-1</math> do       for <math>j \leftarrow 0,  I_{k-1}  - 1</math> do         <math>I_k \leftarrow I_k \cup \{I_{k-1}[j] + in^{k-1}\}</math>       end for     end for     if <math>(n^{d-k} \leq  I_k )</math>       <math>J_k \leftarrow \{0, 1, \dots, n^{d-k} - 1\}</math>     else       for <math>j \leftarrow 0,  J_{k-1}  - 1</math> do         <math>J_k \leftarrow J_k \cup \left\{ \left[ \frac{J_{k-1}[j]}{n} \right] \right\}</math>       end for       while <math> J_k  &lt;  I_k </math> do         <math>J_k \leftarrow J_k \cup \{ \text{RAND}(n^{d-k} - 1) \}</math>       end while     end if     <math>(I_k, J_k) \leftarrow \text{CROSS}(A_k^\alpha(I_k, J_k),  I_k ,  J_k , r_{\max}, \varepsilon)</math>     for <math>l \leftarrow 0,  I_k  - 1</math> do       <math>v_l \leftarrow \text{TO\_POINT}(d, n, k, I_k[l], J_k[l])</math>       <math>v_l \leftarrow \text{OPTIMIZE}(d, e, v_l)</math>       <math>M \leftarrow M \cup \{v_l\}</math>     end for     for <math>l \leftarrow 0,  I_k  - 1</math> do       <math>(i, j) \leftarrow \text{TO\_GRID}(d, n, k, v_l)</math>       <math>I_k \leftarrow I_k \cup \{i\}</math>       <math>J_k \leftarrow J_k \cup \{j\}</math>     end for     <math>\alpha = \min_{m \in M} e(m)</math>   end for </pre>	<p>Шаги слева направо</p> <p>Последовательно рассматриваем матрицы разверток <math>A_k^\alpha</math>.</p> <p>Изменение оптимизируемого функционала в соответствии с <math>\alpha</math>.</p> <p>Соответствующий ему тензор.</p> <p><math>k</math>-я матрица развертки.</p> <p>Построение набора номеров строк <math>I_k</math> матрицы развертки <math>A_k^{(\alpha)}</math> из набора номеров строк <math>I_{k-1}</math> матрицы развертки <math>A_{k-1}^{(\alpha)}</math>.</p> <p>Построение набора номеров столбцов <math>J_k</math>. Если количество столбцов в матрице <math>A_k^\alpha</math> меньше <math> I_k </math>, выбираются все номера столбцов, иначе дополняем имеющиеся номера столбцов с помощью номеров столбцов <math>J_{k-1}</math> матрицы развертки <math>A_{k-1}^\alpha</math> и случайными столбцами, пока количество выбранных столбцов не станет равным количеству выбранных строк</p> <p>Крестовая аппроксимация матрицы <math>A_k^\alpha(I_k, J_k)</math> с точностью <math>\varepsilon</math> и рангом не выше <math>r_{\max}</math></p> <p>Локальная оптимизация узлов интерполяции</p> <p>Вычисление пространственных координат точки, соответствующей элементу <math>(I_k[l], J_k[l])</math> матрицы <math>A_k^\alpha</math>.</p> <p>Локальная оптимизация точки.</p> <p>Добавление в множество локальных оптимумов.</p> <p>Вычисление ближайшего элемента матрицы <math>A_k^\alpha</math> к точке <math>v</math>.</p> <p>Дополнение строковым и столбцовым номером полученного элемента множеств <math>I_k</math> и <math>J_k</math>.</p> <p>Обновление сдвига функционала.</p>
--	---



времени около 90 тысяч структур комплексов белок-ингибитор. Вообще говоря, структура этих комплексов *a priori* не должна соответствовать положению лигандов в белках в реальных организмах или даже просто в тестовых системах *in vitro*, когда наблюдается, например, связывание лиганда-ингибитора с белком-мишенью при добавлении первого в раствор, содержащий белок-мишень и соответствующий субстрат. Это связано не только с тем, что для получения кристалла систему белок-лиганд часто помещают в несвойственные для нее в организме условия, но и с тем, что в кристалле ингибитор фиксирован в определенном положении в активном центре белка, в то время как в организме или в тест-системе имеет место динамическое равновесие в водном растворе белка (P), лиганда-ингибитора (L) и их комплекса (PL), не считая субстрата, взаимодействие которого с белком и должен блокировать ингибитор. Если в кристалле ингибитор находится в одном конкретном положении в белке, которое обычно называют нативным, то в реальности ингибитор движется, колеблясь и перескакивая из одного минимума энергии в другой в результате теплового движения. Таким образом, эффективная программа докинга должна, во всяком случае, быть способной находить нативное положение лиганда-ингибитора или, по крайней мере, некоторое множество положений (их часто называют позами), среди которых присутствует и нативное положение. Необходимость нахождения нескольких вариантов положения лиганда в процессе докинга также обусловлена тем, что нативное положение лиганда может хотя и иметь довольно низкую энергию, но не соответствовать глобальному минимуму скоринг-функции.

В настоящее время функционирование большинства программ докинга опирается на гипотезу о том, что нативному положению лиганда соответствует глобальный минимум энергии системы белок-лиганд. Эта гипотеза соответствует обычному интуитивному представлению о том, что в равновесии система должна находиться в состоянии с наименьшей энергией, т.е. находиться в глобальном минимуме. Более детальное рассмотрение показывает, что в рассматриваемом случае термодинамического равновесия в системе белок-лиганд-комплекс должен достигаться минимум свободной энергии Гиббса, состоящей из энтальпийной и энтропийной составляющих. Однако в большинстве программ докинга этот факт игнорируется и ищется глобальный минимум потенциальной энергии системы белок-лиганд, а энтропийная составляющая свободной энергии либо вообще не учитывается, либо учитывается эмпирически с помощью подгоночных коэффициентов и обучающего набора экспериментальных данных [20]. В настоящей работе мы ограничились поиском нативных положений лигандов и глобального минимума на многомерной поверхности потенциальной энергии системы белок-лиганд, обусловленной силовым полем MMFF94 [17].

При сравнении быстродействия различных алгоритмов докинга естественно их сравнивать по количеству вычислений энергии системы белок-лиганд: чем меньше это число, тем выше быстродействие.

При сравнении принципиально важно, чтобы целевая функция в обоих сравниваемых алгоритмах была одинаковой. Кроме того, надо учесть, что многомерная энергетическая поверхность системы белок-лиганд имеет много локальных минимумов. Учтем также, что в программе докинга SOL вообще не проводится локальная оптимизация лиганда, что затрудняет определение совпадений/несовпадений результатов двух алгоритмов. Поэтому сравнение программ и заложенных в них алгоритмов будем проводить следующим образом.

В качестве целевой функции использована потенциальная энергия лиганда в поле белка-мишени, которая вычисляется в соответствии со следующим алгоритмом. Заранее рассчитывается равномерная сетка потенциалов взаимодействия пробных атомов различных типов с белком-мишенью, созданная программой SOLGRID [2, 3] и охватывающая значительную часть белка, содержащую его активный центр. Обычно это куб со стороной  $22 \text{ \AA}$ , содержащий 101 точку по каждому измерению; центр куба совпадает с геометрическим центром нативного лиганда. В процессе докинга лиганд может занимать произвольное положение в этом кубе, и для каждого такого положения его потенциальная энергия в поле белка-мишени вычисляется в виде суммы вкладов его отдельных атомов, каждый из которых вычисляется путем интерполирования значений энергии пробного атома данного типа по восьми ближайшим точкам сетки потенциалов. Кроме того, учитывается взаимодействие атомов лиганда между собой согласно классическому силовому полю MMFF94. Выбор рассматриваемой целевой функции обусловлен тем, что она обеспечивает высокое быстродействие и широко распространена среди существующих программ докинга [2].

С этой целевой функцией проводился докинг нативного лиганда в соответствующий ему белок-мишень с помощью программы SOL и программы TTDock, причем перед началом каждого докинга нативный лиганд случайным образом раскручивался, так что информация о его нативном положении терялась. Далее, найденные при докинге положения лиганда, соответствующие наиболее отрицательному значению энергии лиганда в поле белка-мишени, использовались в качестве начальных при локальной оптимизации энергии по положениям всех атомов лиганда. При этом энергия системы белок-лиганд вычислялась уже без использования сетки потенциалов в соответствии с силовым полем MMFF94 без уче-

та растворителя. В качестве локальной оптимизации использовался алгоритм L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno), примененный к потенциальной энергии системы белок–лиганд как функции от декартовых координат атомов лиганда. Локальная оптимизация позволяет привести разные найденные положения лиганда в окрестности одного локального минимума в строгое соответствие друг другу. Поскольку в обеих сравниваемых программах докинга использовались эвристические алгоритмы поиска глобального минимума, то каждый процесс докинга повторялся независимо 1000 раз и при анализе результатов учитывалось, насколько часто лиганд попадает в результате докинга в одно и то же положение. Два положения одного и того же лиганда считались идентичными, если среднее квадратичное отклонение координат их соответствующих атомов друг от друга (с учетом симметрии) было меньше  $0.1 \text{ \AA}$ , при этом энергии таких поз различались на величину, меньшую  $0.01 \text{ ккал/моль}$ .

В этих исследованиях использовались семь белков-мишеней и семь соответствующих им нативных лигандов, которые использовались при проведении тестирования CSAR2011 [21]: cdk2\_cs12 (PDB ID 4FKL), cdk2\_cs9 (PDB ID 4FKI), chk1\_8 (PDB ID 4FSW), erk2\_000124 (PDB ID 4FU9), erk2\_000184 (PDB ID 4FV1), erk2\_000633 (PDB ID 4FV8) и urokinase\_7 (PDB ID 4FU9). Лиганды различались как по числу внутренних торсионных степеней свободы (0–9), так и по числу атомов (24–54).

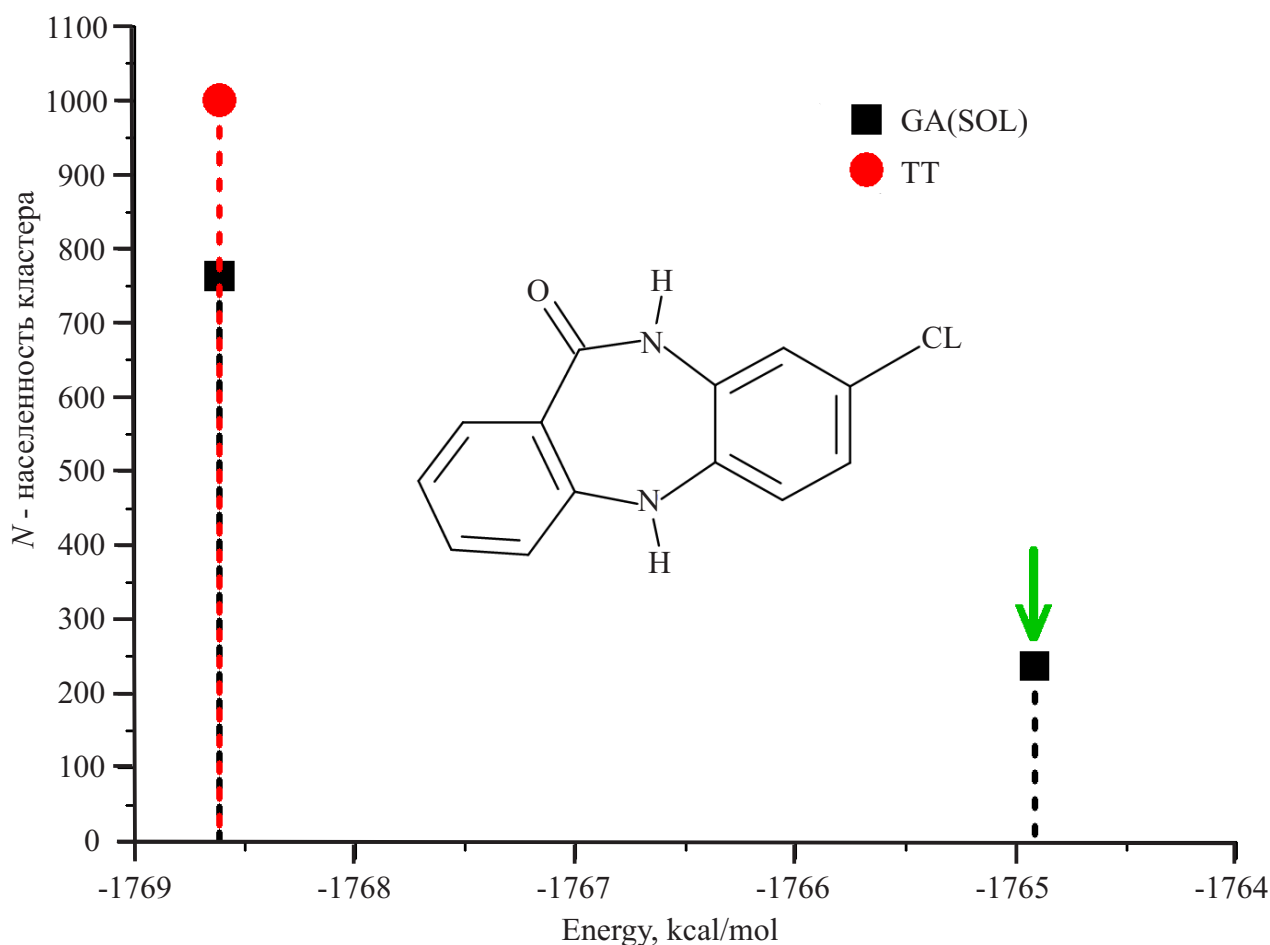


Рис. 1. Результаты докинга жесткого нативного лиганда в белок-мишень *chk1\_8*. На рисунке отображены энергии найденных при докинге положений нативного лиганда с учетом их населенности. По горизонтали отложены энергии соответствующих положений, по вертикали — населенности этих положений; населенность — это количество поз (положений) лиганда в одном кластере; кластер — это множество поз лиганда, среднее квадратичное расстояние между атомами которых меньше  $0.1 \text{ \AA}$  (энергии таких поз различаются менее, чем на  $0.01 \text{ ккал/моль}$ ). Стрелкой отмечено нативное положение лиганда (также после локальной оптимизации). GA(SOL) и TT — результаты докинга программами SOL и TTDock соответственно

На рис. 1 представлены результаты докинга программой SOL и программой TTDock жесткого лиганда (без внутренних вращательных степеней свободы) в белок-мишень *chk1\_8*; по горизонтали отложена энергия комплекса белок–лиганд (после локальной оптимизации по атомам лиганда), которая отличается

от энергии связывания лиганда с белком на константу — энергию свободного белка и свободного лиганда (см. также и на рис. 2 и 3).

Из рис. 1 следует, что программа TTDock не смогла найти нативное положение, а программа SOL нашла его. Однако это не означает, что программа TTDock хуже, чем SOL, так как нативное положение не соответствует наиболее отрицательному минимуму энергии. Глобальный минимум энергии, в свою очередь, уверенно, т.е. с большой населенностью, находится обеими программами, что не удивительно, так как лиганд жесткий и энергетическая поверхность системы белок–лиганд имеет сравнительно небольшое число измерений (всего 6), соответствующих вращениям и трансляциям лиганда как целого.

Совершенно другая ситуация показана на рис. 2.

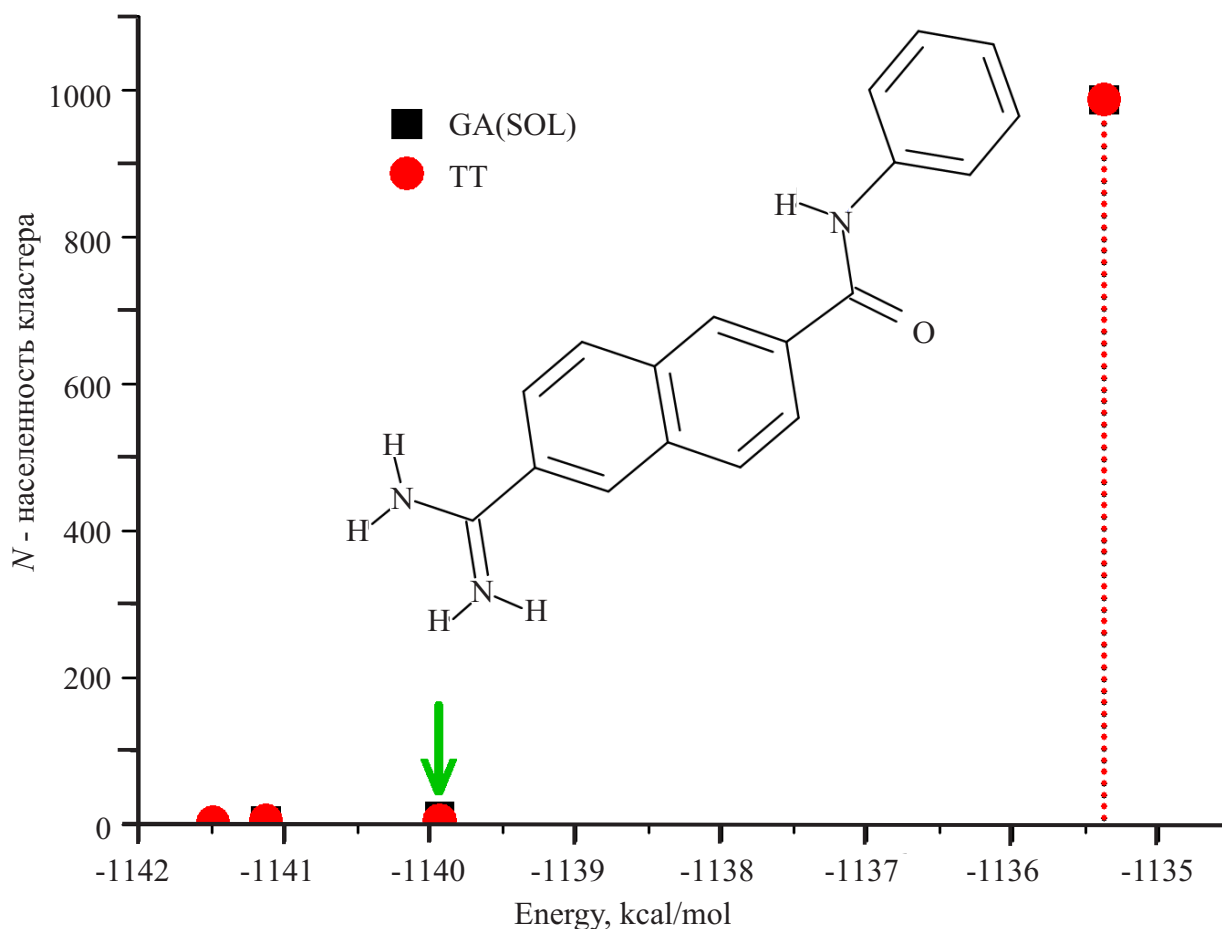


Рис. 2. Результаты докинга гибкого нативного лиганда в белок-мишень урокиназы (urokinase\_7).

Число внутренних вращательных степеней свободы лиганда равно 5. Обозначения такие же, как и на рис. 1

Из рис. 2 следует, что нативное положение лиганда находится обеими программами, хотя и с большим трудом — населенность соответствующего кластера очень небольшая. Видно также, что энергия кластера с очень высокой населенностью может не соответствовать ни глобальному минимуму энергии, ни энергии нативного положения лиганда. На рис. 1 количество кластеров, полученных при докинге программами SOL и TTDock, равно 2 и 1, а на рис. 2 количество кластеров 3 и 4 соответственно.

Из рис. 3 видно, что количество кластеров может быть и довольно большим даже для сравнительно жесткого лиганда (всего 4 внутренних торсиона). На исследованных нами белках и лигандах большая кластеризация, т.е. большой разброс найденных положений лиганда, наблюдается у программы SOL по сравнению с программой TTDock.

Приведенные примеры иллюстрируют следующие обнаруженные нами закономерности:

- нативное положение лиганда не обязательно соответствует глобальному минимуму целевой функции;
- среди объединенного множества найденных минимумов (SOL+TTDock) всегда есть оптимизированное нативное положение, но программа SOL не нашла нативное положение в двух случаях, а программа

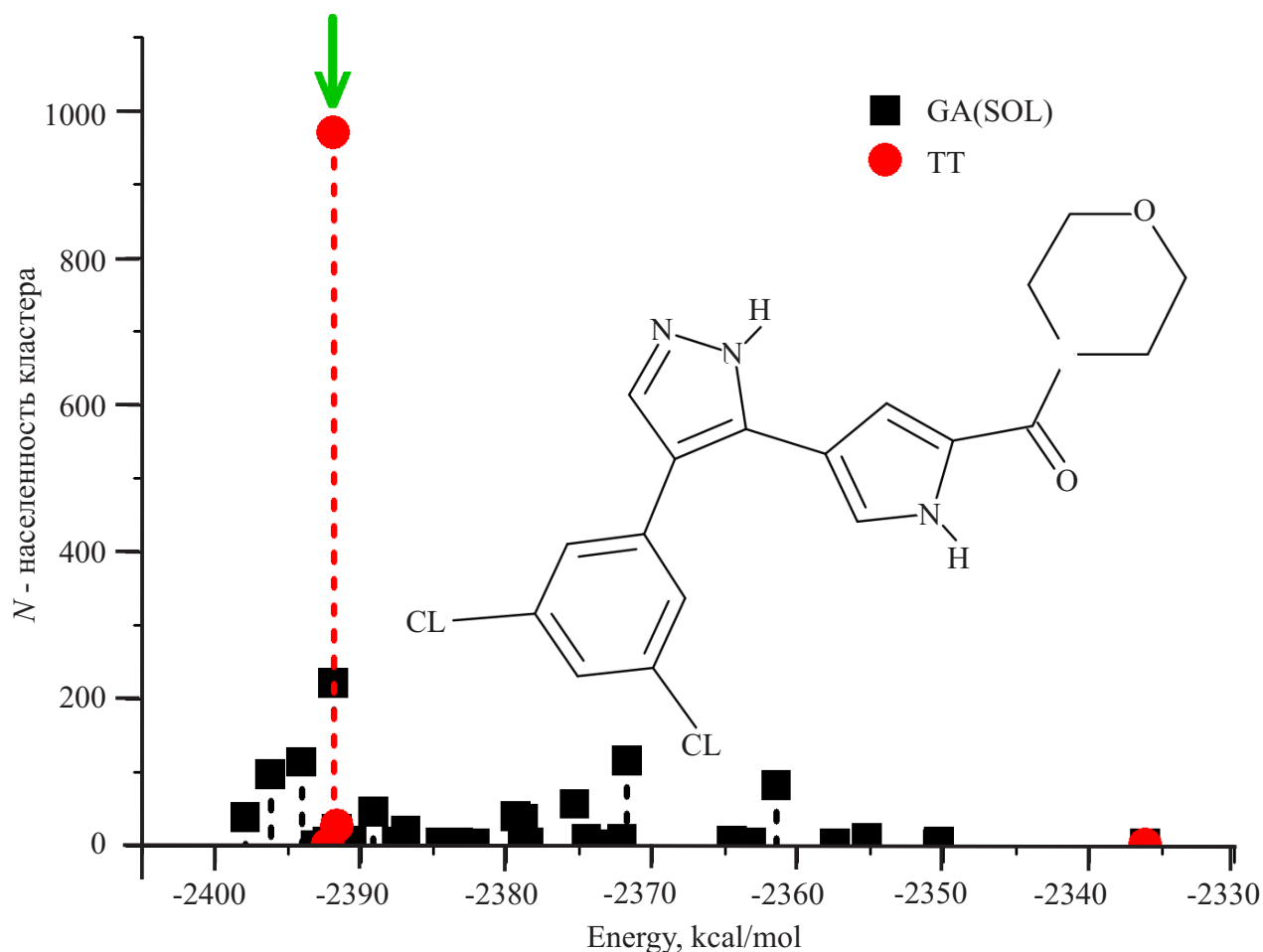


Рис. 3. Результаты докинга нативного лиганда в белок-мишень erk2\_000124. Число внутренних вращательных степеней свободы лиганда равно 4. Обозначения такие же, как на рис. 1

TTDock не нашла нативное положение в одном случае;

- нет заведомо лучшего метода: наименьшие найденные энергии при помощи SOL и TTDock примерно одинаковы;
- SOL дает больший разброс результатов, чем TTDock.

Таким образом, можно заключить, что обе программы докинга SOL и TTDock с учетом большого числа независимых запусков для накопления статистики имеют примерно одинаковую надежность как нахождения глобального минимума системы белок-лиганд, так и нативного положения лиганда. Что же касается быстродействия, то во всех исследованных случаях докинг с помощью программы TTDock оказался быстрее докинга с помощью программы SOL: количество вычислений энергии системы белок-лиганд в первом случае было в 10 раз меньше, чем во втором.

**4. Заключение.** В настоящей статье представлен алгоритм докинга, использующий методы, основанные на ТТ-разложениях тензоров высокой размерности, и его программная реализация TTDock.

Приведено сравнение эффективности и быстродействия разработанной программы TTDock и программы докинга SOL. Показано, что по числу операций, которые необходимо выполнить для решения задачи докинга (нахождения глобального минимума системы белок-лиганд) программа докинга TTDock на один порядок быстрее программы докинга SOL. Столь существенное увеличение быстродействия докинга открывает возможность реализации прямого бессеточного докинга на основе ТТ-разложений, а в дальнейшем и докинга с учетом подвижности атомов активного центра белка-мишени, что в конечном итоге должно привести к более точному предсказанию констант ингибирования и увеличению эффективности применения докинга для разработки новых лекарств.

Проведенные исследования показали, что обе программы докинга SOL и TTDock могут находить нативное положение лиганда или положение с энергией ниже, чем энергия локально оптимизированного нативного положения лиганда. Однако программа TTDock находит подобное положение чаще, чем программа SOL, т.е. разброс результатов у программы SOL заметно больше.

Если объединить решения, найденные обеими программами, то для всех исследованных случаев среди них окажется и нативное положение лиганда. Приведенные примеры показывают, что нативное положение может не совпадать с положением лиганда, соответствующим наименьшей энергии, и происходит это не так уж редко. Таким образом, ни нативное положение лиганда, ни положение, соответствующее наименьшему значению энергии, не являются во многих случаях выделенными; около них по энергии может быть много локальных энергетических минимумов, а положения этих минимумов могут заметно отличаться друг от друга, от глобального минимума и от нативного положения лиганда. Отсюда следует, что основная задача докинга заключается в нахождении не только глобального минимума энергии системы белок–лиганд, но и нахождении всех расположенных рядом с ним по энергии (в диапазоне нескольких кТ) локальных минимумов, в которых эта система может находиться из-за тепловых флуктуаций.

Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 11.03.2013 г. № 14.514.11.4070 в рамках ФЦП “Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы”.

#### СПИСОК ЛИТЕРАТУРЫ

1. Садовничий В.А., Сулимов В.Б. Суперкомпьютерные технологии в медицине // Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред. В.А. Садовничего, Г.И. Савина, Вл.В. Воеводина. М: Изд-во Моск. ун-та, 2009. 16–23.
2. Романов А.Н., Кондакова О.А., Григорьев Ф.В., Сулимов А.В., Луцкекина С.В., Мартынов Я.Б., Сулимов В.Б. Компьютерный дизайн лекарственных средств: программа докинга SOL // Вычислительные методы и программирование. 2008. **9**. 213–233.
3. Офёркин И.В., Сулимов А.В., Кондакова О.А., Сулимов В.Б. Реализация поддержки параллельных вычислений в программах докинга SOLGRID и SOL // Вычислительные методы и программирование. 2011. **12**. 9–23.
4. Mobley D.L., Dill K.A. Binding of small-molecule ligands to proteins: “what you see” is not always “what you get” // Structure. 2009. **17**, N 4. 489–498.
5. Kolda T.G., Bader B.W. Tensor decompositions and applications // SIAM Review. 2009. **51**, N 3. 455–500.
6. Oseledets I.V., Savostyanov D.V., Tyrtyshnikov E.E. Tucker dimensionality reduction of three-dimensional arrays in linear time // SIAM J. Matrix Anal. Appl. 2008. **30**, N 3. 939–956.
7. Oseledets I.V., Tyrtyshnikov E.E. Breaking the curse of dimensionality, or how to use SVD in many dimensions // SIAM J. Sci. Comput. 2009. **31**, N 5. 3744–3759.
8. Oseledets I.V. Tensor-train decomposition // SIAM J. Sci. Comput. 2011. **33**, N 5. 2295–2317.
9. Oseledets I.V., Tyrtyshnikov E.E. TT-cross approximation for multidimensional arrays // Linear Algebra Appl. 2010. **432**, N 1. 70–88.
10. Goreinov S.A., Tyrtyshnikov E.E., Zamarashkin N.L. A theory of pseudo-skeleton approximations // Linear Algebra Appl. 1997. **261**, N 1–3. 1–21.
11. Tyrtyshnikov E.E. Incomplete cross approximation in the mosaic-skeleton method // Computing. 2000. **64**, N 4. 367–380.
12. Goreinov S.A., Tyrtyshnikov E.E. The maximal-volume concept in approximation by low-rank matrices // Contemporary Mathematics. 2001. **208**. 47–51.
13. Goreinov S.A., Oseledets I.V., Savostyanov D.V., Tyrtyshnikov E.E., Zamarashkin N.L. How to find a good submatrix // Matrix Methods: Theory, Algorithms, Applications / Edited by V. Olshevsky and E. Tyrtyshnikov. Hackensack: World Scientific, 2010. 247–256.
14. Huey R., Morris G.M., Olson A.J., Goodsell D.S. A semiempirical free energy force field with charge-based desolvation // J. Computational Chemistry. 2007. **28**, N 6. 1145–1152.
15. Verdonk M.L., Cole J.C., Hartshorn M.J., Murray C.W., Taylor R.D. Improved protein–ligand docking using GOLD // Proteins: Structure, Function, and Bioinformatics. 2003. **52**, N 4. 609–623.
16. Thomsen R., Christensen M.H. MolDock: a new technique for high-accuracy molecular docking // J. Medicinal Chemistry. 2006. **49**, N 11. 3315–3321.
17. Halgren T.A. Merck molecular force field // J. Computational Chemistry. 1996. **17**, N 5/6. 490–519.
18. Romanov A.N., Jabin S.N., Martynov Y.B., Sulimov A.V., Grigoriev F.V., Sulimov V.B. Surface generalized Born method: a simple, fast, and precise implicit solvent model beyond the Coulomb approximation // J. Physical Chemistry A. 2004. **108**, N 43. 9323–9327.
19. RCSB protein data bank (<http://www.rcsb.org/pdb/home/home.do>).
20. Alvarez J., Shoichet B. Virtual screening in drug discovery. Vol. 1. Boca Raton: CRC Press, 2005.
21. CSAR 2011–2012 Benchmark Exercise (<http://www.csardock.org/>).

Поступила в редакцию  
09.05.2013