

# 基于 CPU 和 GPU 异构的蛋白质分子半柔性对接算法优化

陆旭峰<sup>1</sup> 陆振宇<sup>1</sup> 梅向东<sup>2</sup> 孔 韧<sup>1</sup> 常 珊<sup>1</sup>

(1. 江苏理工学院电气信息工程学院生物信息与医药工程研究所, 常州, 213001; 2. 江苏赞奇科技股份有限公司, 常州, 213022)

**摘 要:** 分子对接是预测蛋白质复合物的有效手段。对于分子对接算法的优化旨在加速分子对接效率, 降低计算成本, 以及充分发挥计算资源的利用率。本文主要采用 3 个方案对半柔性对接算法进行优化: (1) 方案一在 CPU 端进行优化; (2) 方案二在方案一的基础上, 利用 CUFFT 的移植工具 CUFFTW 为方案一提供部分 GPU 并行接口; (3) 方案三利用 GPU 并行架构, 通过 CPU 和 GPU 的协同处理, 利用纯并行计算接口进行优化。3 种方案对 PDB code 分别为 1PEE, 1B6C, 4HX3 和 2SNI 的测试蛋白进行结合态和自由态的对接, 求得的最小均方根偏差  $L_{\text{RMSD}}$  小于  $5 \text{ \AA}$ , 满足了复合物结构预测竞赛要求的中等精度结构标准, 验证了对接结果的正确性。最后在保证结果正确性的前提下, 测试了不同蛋白在不同方案下的运行速率; 在保证不同蛋白对接效率相同的前提下, 以 1PPE 为例, 比较了不同方案下的对接速率。实验结果表明在同等旋转步长并保证程序运行结果正确性的前提下, 最终的优化效果可提速近 10 倍, 有效改进了半柔性对接算法的运行速率。

**关键词:** 蛋白质; 半柔性; 分子对接; CPU; GPU

**中图分类号:** TP301      **文献标志码:** A

## Optimization of Semi Flexible Docking Algorithm for Protein Molecules Based on CPU and GPU Heterogeneous

Lu Xufeng<sup>1</sup>, Lu Zhenyu<sup>1</sup>, Mei Xiangdong<sup>2</sup>, Kong Ren<sup>1</sup>, Chang Shan<sup>1</sup>

(1. Institute of Bioinformatics and Medical Engineering, School of Electrical and Information Engineering, Jiangsu University of Technology, Changzhou, 213001, China; 2. Cudatec Development Co., Ltd, Changzhou, 213022, China)

**Abstract:** Molecular docking is an effective method to predict protein complexes. The optimization of molecular docking algorithm is aimed at accelerating the efficiency of molecular docking, reducing the computational cost, and making full use of computer resources. This paper mainly optimizes the semi flexible docking algorithm using three schemes: (1) The first scheme is optimized on the CPU side. (2) The second scheme, on the base of the first scheme, uses the CUFFT transplantation tool CUFFTW to provide partial GPU parallel interface for the solution. (3) The third scheme is based on the compute unified device architecture (CUDA) parallel structure, which is achieved by a pure CUDA interface and the coordi-

**基金项目:** NSFC-广东联合基金(第二期)超级计算科学应用研究专项资助项目; 国家自然科学基金(11647146, 81603152)资助项目; 江苏省六大人才高峰(2016-XYDXXJS-020)资助项目; 江苏省产学研前瞻(BY2016030-06)资助项目; 江苏省研究生科研与实践创新计划(SJCX17\_0747)资助项目; 江苏省研究生科研与实践创新计划(SJCX17\_0748)资助项目。

**收稿日期:** 2018-02-09; **修订日期:** 2018-05-09

nated operation of CPU and GPU. The three schemes use four different proteins to test the bound docking and unbound docking. The proteins' PDB code is 1PEE, 1B6C, 4HX3 and 2SNI, respectively. The correctness of test is verified based on the minimum root mean square deviation  $L_{\text{RMSD}}$ , and its value is less than 5 Å, which satisfies the CAPRI medium precision structural standard. Then, under the premise of ensuring the correctness of the results, the running speed of different proteins under different schemes is tested. Finally, under the premise of ensuring the same efficiency of different protein docking, we take 1PPE as the final object to point out the rate of docking under different schemes. Experimental results show that under the same rotation step and the correctness of the program, the final optimization effect can increase the speed of the algorithm by nearly 10 times, thus greatly improving the speed of the algorithm.

**Key words:** protein; semi flexible; molecular docking; CPU; GPU

## 引 言

分子对接是用来预测分子之间相互作用和相互识别的模式,而蛋白质分子对接是预测蛋白质受体和蛋白质配体的结合匹配模式<sup>[1]</sup>。通常,生物学实验是探究分子之间的相互作用的直接手段。但通过实验直接研究蛋白质相互作用并不容易,获得蛋白质复合物晶体的生物实验条件比较严苛,并且实验的成本也非常昂贵,这使理论模拟方法得到了越来越多的重视和应用。分子对接方法在研究分子间相互作用领域有着广泛的应用,可减少单纯实验带来的种种困难,为实验者们提供重要的理论依据。

分子对接主要分为3种,分别是刚性对接(Rigid docking),半柔性对接(Semi flexible docking)和柔性对接(Flexible docking)<sup>[2-3]</sup>。柔性对接指的是在对接过程中,受体和配体都发生构象的改变,这种对接在于精度上得到了很好的保证,但是随之产生的负面效应是计算量的大幅增加,而刚性对接则恰恰相反。本文主要采用的是半柔性对接,即在保证受体构象不发生变化的情况下,配体的构象发生一定的改变,在计算效率和精度上都能达到相应的要求。

现在被广泛使用的分子对接软件主要有:AutoDock, Molegro Virtual Docker, 3D\_Dock, DOCK 以及 FlexX 等<sup>[4]</sup>。其中,3D\_Dock 程序具有很大的优越性<sup>[5]</sup>,能够快速进行全局对接和复合物结构数据的筛选。但是,随着对接数据量的不断增大以及计算机硬件设备及处理能力的不断更新,传统的3D\_Dock 程序已经不能胜任目前的情况。特别是针对半柔性对接计算量比较大,原有的计算方法和并行策略都明显需要改进。

NVIDIA 公司发布的统一计算设备架构(Compute unified device architecture, CUDA)<sup>[6-7]</sup>,让 GPU 作为计算单元得到了强有力的开发。GPU 作为图形处理器具有超强的浮点运算能力,而将 GPU 从传统的图形处理器变为计算单元来使用,最大化地利用其浮点运算能力,有可能解决上述分子对接面临的问题。如何将 GPU 和 3D\_Dock 程序结合起来是本文的研究重点。

本文在原有 3D\_Dock 程序基础上进行改进。首先,在 CPU 上,基于 FFTW3 对原有的程序进行算法升级和程序改进。然后,在 GPU 上,基于 CUFFTW 和 CUFFT 对改进后的程序进行再升级。通过 3 种方法对程序进行改进和算法升级,用改进程序对 PDB code 为 1PPE 的测试蛋白进行结合态和自由态对接,结果验证了程序的正确性并给出了改进后程序的加速比。

## 1 3D\_Dock 主要流程

分子对接首先要对受体和配体分子结合模式进行全空间搜索<sup>[8-9]</sup>,但是,蛋白质的自身特点决定了空间搜索的困难,即蛋白质对接所产生的复合物数量十分庞大。传统的用于分子对接的空间搜索算法主要有遗传算法、模拟退火算法、快速傅里叶变换算法(Fast Fourier transform, FFT)、片段生长法以及

蒙特卡罗算法等。3D\_Dock程序主要采用FFT算法来进行复合物构象的计算,只要所需评价函数满足 $\sum_{ij} p_i q_j$ 的形式,就可以通过FFT算法进行程序的加速,提高采样速度。其主要理论依据在于通过离散函数表现分子特性,具体的流程如下:

假设需要进行对接的受体和配体分子的特性,以及通过对接得到的复合物的特性都通过格点表现出来,则可以通过格点特性来求解特性匹配值。匹配值计算公式如下

$$\vec{c}_{\alpha,\beta,\gamma} = \sum_{l=1}^N \sum_{m=1}^N \sum_{n=1}^N \vec{a}_{l,m,n} \cdot \vec{b}_{l+\alpha,m+\beta,n+\gamma} \quad (1)$$

式中: $\vec{a}$ 为受体投影离散函数; $\vec{b}$ 为配体的投影离散函数; $N$ 为格点数; $\alpha,\beta,\gamma$ 分别为配体质心在3个坐标轴上偏离受体质心的格点步数。以下为快速傅里叶变换具体流程:

**步骤1** 确定受体分子和配体分子的投影离散函数 $\vec{a}_{l,m,n}, \vec{b}_{l,m,n}$ ,其过程主要是通过两者的原子坐标进行确定和计算。

**步骤2** 分别对投影离散函数 $\vec{a}, \vec{b}$ 进行时域到频域的变化,即进行正向傅里叶变换,其中需要注意的是对受体需要在正向快速傅里叶变换的基础上继续求取复共轭。

**步骤3** 求频域的点乘。

**步骤4** 根据步骤3得到了所需的复合物结构数据 $C_{\alpha,\beta,\gamma}$ ,但是得到的值为频域下的值,因此,下一步需要进行的是傅里叶逆变换。

主要计算公式如下

$$A_{\alpha,\beta,\gamma}^* = \{\text{FFT}(\vec{a}_{l,m,n})\}^* \quad (2)$$

$$B_{\alpha,\beta,\gamma} = \text{FFT}(\vec{b}_{l,m,n}) \quad (3)$$

$$C_{\alpha,\beta,\gamma} = A_{\alpha,\beta,\gamma}^* \cdot B_{\alpha,\beta,\gamma} \quad (4)$$

$$\vec{c}_{\alpha,\beta,\gamma} = \text{IFFT}(C_{\alpha,\beta,\gamma}) \quad (5)$$

式(2)为对 $\vec{a}$ 进行傅里叶变换并求复共轭,式(3)为对 $\vec{b}$ 进行傅里叶变换,式(4)为求两者点乘,式(5)为求傅里叶逆变换。

经过式(5)计算后得到复合物的相关特性匹配值,根据排序筛选,取得该轮计算的极小值并进行保存,至此便完成了一轮计算。接下来通过改变配体的不同角度继续进行对接,重复以上步骤,得到一系列对接复合物结构。需要特别注意的是,因为是半柔性对接,即受体在构象上不发生变化,因此受体只需要进行一次傅里叶变换即可。完成所有对接情况后,对保留下来的数据继续进行排序,得到打分项最高的复合物结构。

## 2 优化方案介绍

### 2.1 基于 FFTW 的优化方案一

3D\_Dock的核心程序是FFTW。FFTW主要有两个经典版本FFTW2和FFTW3,但是FFTW3并不兼容FFTW2,而原程序的版本为FFTW2<sup>[10-11]</sup>。因此,需要有在CPU上优化程序的方案,即将原来FFTW2的函数全部升级为FFTW3的函数。优化方案一基于CPU,将原程序的FFTW2进行升级为FFTW3,其主要流程如下:

**步骤1** 准备工作:主要涉及内存分配、默认参数的设置、角度计算、原子带电量设定、离散后网格划分、动静态分子结构的读入,以及建立傅里叶变换和逆变换策略。

**步骤2** 将静态分子进行平面化和离散化,动态分子先采用分子动力学模拟生成3种构象,然后进行离散化。

**步骤3** 分别将静态分子和动态分子得到的网格数据进行傅里叶变换,并进行点乘计算,得到复合物投影离散函数,并对其进行傅里叶逆变换,得到复合物相关函数。

**步骤 4** 对得到的复合物相关函数进行匹配因子计算,并进行打分排序,筛选出最好的 3 个结果进行保留。

**步骤 5** 将动态分子进行旋转,重复步骤 3,4,直到所有旋转结束。

**步骤 6** 将得到的所有结果进行排序,并写入文件中,上限为 10 000 个数据。

**步骤 7** 按上述步骤完成受体与另外两个动态分子构象的对接测试。

## 2.2 基于 CUFFT 的优化方案二

随着 GPU 技术的不断革新,NVIDIA 公司提出了 CUDA 架构,同时也针对傅里叶变换发布了两种加速方案,一种是 CUFFT 加速,另一种是 CUFFT 加速。

CUFFT 和 CUFFT 的区别主要在于,CUFFT 是针对 FFTW 库开发的,可以最快速、最便捷地为使用 FFTW 库(方案一采用的加速方法)的用户带来 GPU 的加速效果,即只要原程序采用的是 FFTW 库,那么通过 CUFFT 可以直接在 GPU 端进行运行,不需要太多的改动。而 CUFFT 则不同,它是针对所有傅里叶变换开发的,是一种完全的 CUDA 接口。

优化方案二在优化方案一的基础上,采用 CUFFT 通过部分 CUDA GPU 接口进行加速。换言之,方案二是将原来在 CPU 端运行的程序移植到 GPU 端运行,通过 CUDA GPU 架构来对方案一进行再加速。具体流程与方案一相同,只是运行环境发生了改变。

## 2.3 基于 CUFFT 的优化方案三

优化方案一虽然程序确实得到了加速,但是加速效果并不理想;优化方案二专门针对 FFTW 库开发,不是采用纯 CUDA GPU 接口,若换成其他函数库则无法使用。因此,考虑到 CUFFT 版本的加速效果,本文又提出优化方案三,即在 GPU 端进行纯 CUDA 接口傅里叶变换,将耗时的快速傅里叶变换运算和复合物点乘计算移植到 GPU 端,将所得结果返回 CPU 进行筛选排序处理,并利用 GPU 的高速浮点运算能力加速程序的运行<sup>[12-13]</sup>。具体运输步骤如下:

**步骤 1** 根据所做的工作制定一个计划。因为本文傅里叶变换是在三维空间进行的,因此需要调用的函数是 `cufftPlan3d`。其他低维情况可以参考 CUFFT 的官方文档做相应变动。

**步骤 2** 调用相关函数开始执行计划。因为本文所做的是双精度的傅里叶变换和傅里叶逆变换,所以需要调用的函数分别是 `cufftExecR2C()` 和 `cufftExecC2R()`。

**步骤 3** 调用函数 `cufftDestroy()` 释放资源并销毁计划。

以一种动态分子构象计算为例,基于 CUDA GPU 的 3D\_Dock 程序的具体流程如图 1 所示。

## 3 分子对接实例及结果分析

### 3.1 对接实例

为了测试 3D\_Dock 两种优化方案的有效性以及计算精度,在采用 3D\_Dock 默认对接参数的基础上,将程序应用于 PDB code 为 1PEE,1B6C,4HX3 和 2SNI 的结合态分子对接和自由态分子对接(图 2),求取结合态和自由态模式下的前 10 名复合物的最小均方根

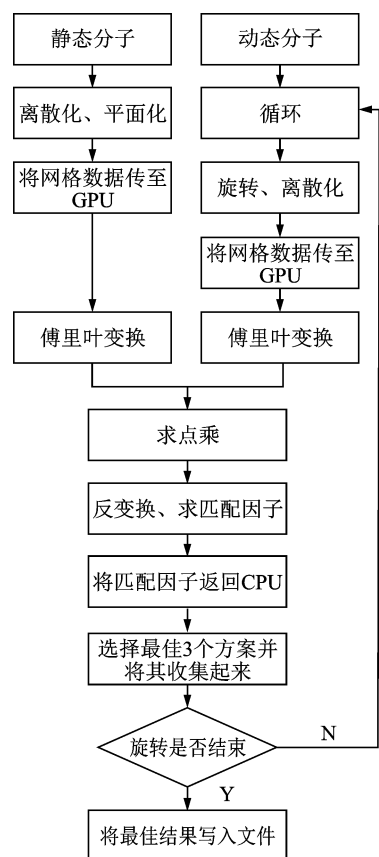


图 1 基于 GPU 的分子对接流程图  
Fig. 1 Flow chart of molecular docking based on GPU

偏差  $L_{\text{RMSD}}$  值<sup>[14-15]</sup>,同时展示4个体系复合物预测的结构图,比较对接结果与原晶体结构的差异,验证结果的正确性,如表1所示。

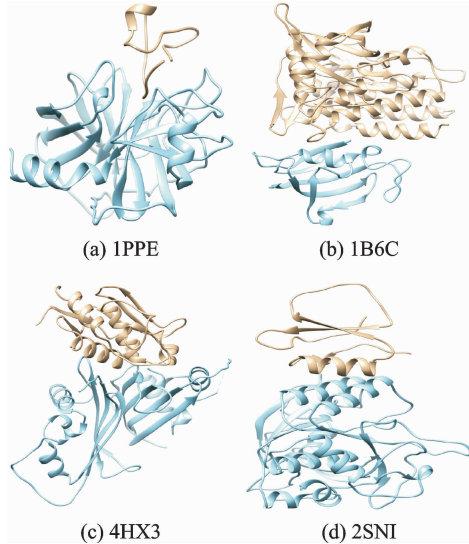


图2 PDB code分别为1PPE,1B6C,4HX3和2SNI的蛋白质分子结构(淡黄色为配体,蓝色为受体)

Fig. 2 Molecular structures of 1PPE, 1B6C, 4HX3 and 2SNI (PDB code) (Pale yellow shows the ligand and blue the receptor)

评价方式采用蛋白质复合物结构预测竞赛(Critical assessment of predicted interactions, CAPRI<sup>[16-17]</sup>)组委会的指标。在CAPRI实验中,正确的预测结果分为高精度结构、中等精度结构和可接受结构<sup>[18-19]</sup>。根据 $L_{\text{RMSD}}$ 划分情况为:当 $L_{\text{RMSD}}$ 小于1 Å时为高精度结构;当 $L_{\text{RMSD}}$ 小于5 Å时为中等精度结构;当 $L_{\text{RMSD}}$ 小于10 Å时为可接受结构。 $L_{\text{RMSD}}$ 表示预测的配体构象与晶体结构中配体结构的均方根偏差。本文选取的标准为 $L_{\text{RMSD}}$ 小于5 Å,即为中等精度下为近天然结构。具体数据来源于CAPRI官方网站:<http://www.ebi.ac.uk/msd-srv/capri/>。

根据表1可以看出,结合态结构的 $L_{\text{RMSD}}$ 值要比自由态结构的 $L_{\text{RMSD}}$ 值小。其主要原因是自由态对接主链结构有明显变化,因此自由态的 $L_{\text{RMSD}}$ 值通常要比结合态对接的值大。而结合态结构是直接由复合物晶体结构拆分而来,因此在对接中更容易找到接近晶体结构的复合物构象。图3是4个PDB结果在结合态结构最低值时(根据表1结果)对应的对接结果

表1 对接4个体系中具有最小 $L_{\text{RMSD}}$ 值的前10名结合态和自由态结果

Tab. 1 The first ten bound and unbound results of docking instance with the minimum  $L_{\text{RMSD}}$  value

PDB 编号	受体分子	配体分子	结合态 $L_{\text{RMSD}}/\text{Å}$	自由态 $L_{\text{RMSD}}/\text{Å}$
1PPE	Bovine trypsin	CMTI-1 squash inhibitor	2.09	3.77
1B6C	FK506-BINDING	TGF-B superfamily receptor type I	2.18	2.24
4HX3	Neutral proteinase	Extracellular small neutral pro-inhibitor ScNPI	0.93	3.73
2SNI	Subtilisin	Chymotrypsin inhibitor 2	1.17	3.43

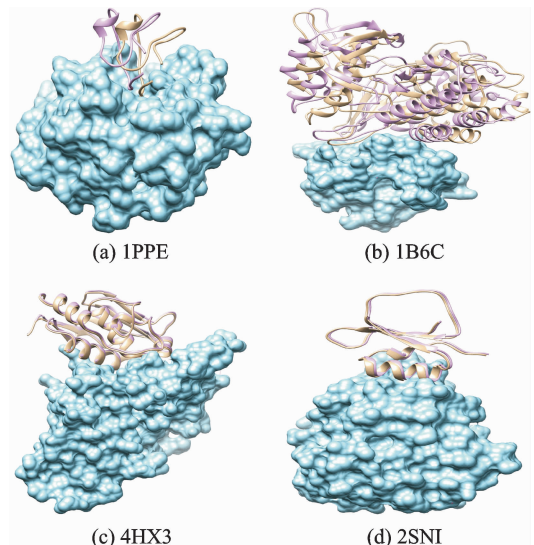


图3 4个PDB结合态结果在 $L_{\text{RMSD}}$ 最低值时与原晶体结构对比图

Fig. 3 Comparison of four PDB bound results with the minimum values of  $L_{\text{RMSD}}$  and the original crystal structure

与原晶体结构的对比图,从图中可以确认对接结果的正确性。

### 3.2 性能分析

为了测试两种方案对于程序的优化效果,对上述 4 种蛋白均进行了测试,并且加速比基本一致。为便于讨论,选取受体 Bovine trypsin 和配体 CMTI-1 squash inhibitor(PDB code: 1PPE)的结合态和自由态对接作为优化测试体系。在保持网格数一定的前提下,通过改变旋转角度步长来比较优化的效果,分别取旋转角度为  $12^\circ, 15^\circ, 18^\circ$  进行测试。表 2, 3 为 3 种方案结合态和自由态结构的测试结果,每个测试结果都是配体 3 种构象的平均值,表中原方案在 CPU 上运行,采用 FFTW2 函数库。

表 2 基于结合态的 4 种方案运行时间

Tab. 2 Running time of four optimization schemes based on bound structure

旋转角度 步长/ $(^\circ)$	结合态分子对接方案运行时间/s			
	原方案	优化方案 一	优化方案 二	优化方案 三
12	9 805.99	3 553.28	1 207.79	907.01
15	4 909.90	1 807.44	625.58	476.89
18	2 820.54	1 047.82	365.27	290.95

表 3 基于自由态的 4 种方案运行时间

Tab. 3 Running time of four optimization schemes based on unbound structure

旋转角度 步长/ $(^\circ)$	自由态分子对接方案运行时间/s			
	原方案	优化方案 一	优化方案 二	优化方案 三
12	8 842.64	3 150.29	1 058.62	860.66
15	4 717.49	1 691.28	580.51	474.74
18	2 610.71	949.71	349.40	288.23

表 2 中优化方案一较原方案在运行时间上有较大幅度的降低。这是由于原方案采用的快速傅里叶变换函数库为 FFTW2,而优化方案一采用的函数库为 FFTW3。版本 2 的傅里叶变换主要有 `fftw.h` 和 `rfftw.h` 两个头文件,分别代表傅里叶变换和傅里叶逆变换;而版本 3 的傅里叶变换头文件只有 1 个头文件 `fftw3.h`,即无论是正变换还是逆变换都集中于该头文件。此外,在版本 2 的傅里叶变换中,原来的实数被双精度浮点型替换,使得精度得到了较大的提高。而且原复数的实部和虚部是在结构体中定义,而在版本 3 中直接进行宏定义,这样使用起来更加方便和快捷。同时,FFTW2 和 FFTW3 的差别还在于策略的生成与执行。FFTW3 将傅里叶变换和反变换的重点从策略的制定转移到策略的执行。如果把计算都放在了策略的生成上面,那么每次生成策略都需要消耗一定的时间,而分子对接工作需要多次傅里叶变换计算,因此采用 FFTW2 函数库运行时间会大幅增加,而采用 FFTW3 则可以降低运行时间。优化方案二、三分别是在方案一的基础上进行的再升级,利用 CUFFT 的两个子库使得部分程序能够在 GPU 端进行运算,从而降低了程序的运行时间,较之方案一,两者的提速比分别达到 3 倍和 10 倍。

由表 3 可知,自由态结构和结合态结构一样,在改进后的方案中速度都得到了提高,并且提高幅度相差不大。但是自由态结构的运行时间普遍比结合态结构的运行时间少,这主要是因为部分结合态结构计算所需的格点数要比自由态结构多<sup>[20]</sup>,因此程序运行消耗的时间也相应会有所提高,但是增加的时间并不多,这也一定程度上证实了程序的正确性,以及优化方法的正确性。

图 4, 5 是结合态结构和自由态结构中 4 种方案在旋转角度为  $12^\circ, 15^\circ, 18^\circ$  时的加速比结果,图中所求得的加速比都是配体 3 种构象测试结果的平均值。

通过表 2, 3 及图 4, 5 可以得到两方面结论。首先,当旋转步长不断增大时,程序的运行速率得到了提高,这主要是因为当旋转步长增加时,需要旋转采样的次数随之减少,这样所需要的计算量便大幅降低,因此程序运行时间自然减少。其次,如果旋转采样的次数减少,所需要使用的傅里叶变换次数便随之减少,而本文所进行的优化大部分是针对傅里叶变换进行的,一旦傅里叶变换计算次数减少,那么总体的加速比自然会下降。相反,如果旋转步长降低,程序的优化效果则会得到较大的提高,原因在于旋转次数增加,对接采样的角度更加细化,使用快速傅里叶变换的计算次数也会增加。以上结论可以证明本文的优化方案有效,通过对最主要的傅里叶变换算法进行提速,使得程序运行效率得到较大的提高。

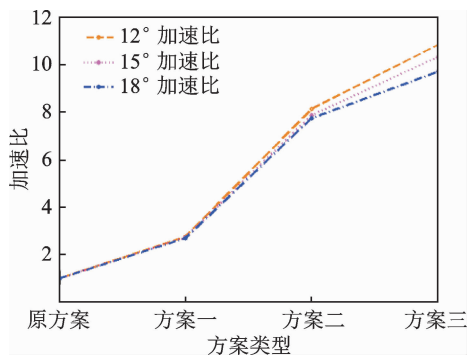


图4 使用4种不同方案时结合态结构加速比对比图

Fig. 4 Accelerated line chart of bound structures using four schemes

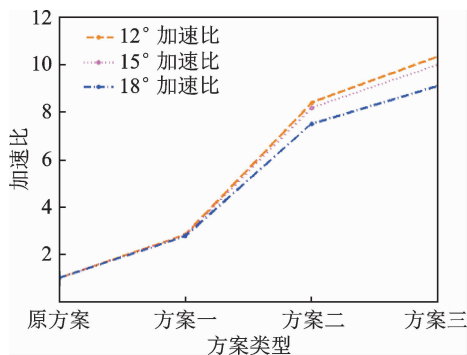


图5 使用4种不同方案时自由态结构加速比对比图

Fig. 5 Accelerated line chart of unbound structures using four schemes

## 4 结束语

本文主要针对3D\_Dock中傅里叶变换提出了3种不同程度的优化方案,分别是基于CPU的优化和两种基于GPU的优化。从实验结果可以证明较之原程序而言,3种优化方案都提高了程序的运行速度,特别是基于GPU的方案,并行计算的优化更大程度上提高了程序的运行速度。这3种优化方案对于蛋白质结构预测领域<sup>[21]</sup>有着重要的应用,可以降低计算成本,提高预测效率。

## 参考文献:

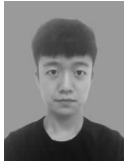
- [1] Ritchie D W. Recent progress and future directions in protein-protein docking[J]. *Current Protein & Peptide Science*, 2008, 9(1): 1-15.
- [2] 常珊, 孔韧, 李春华, 等. 基于MPI的分子对接并行算法[J]. *计算物理*, 2008, 25(2): 241-246.  
Chang Shan, Kong Ren, Li Chunhua, et al. Parallel algorithm for molecular docking based on MPI[J]. *Chinese Journal of Computational Physics*, 2008, 25(2): 241-246.
- [3] 王存新, 常珊, 龚新奇, 等. 蛋白质-蛋白质分子对接中打分函数研究进展[J]. *物理化学学报*, 2012, 28(4): 751-758.  
Wang Cunxin, Chang Shan, Gong Xinqi, et al. Progress in the scoring functions of protein-protein docking[J]. *Acta Physico-Chimica Sinica*, 2012, 28(4): 751-758.
- [4] 段爱霞, 陈晶, 刘宏德, 等. 分子对接方法的应用与发展[J]. *分析科学学报*, 2009, 25(4): 473-477.  
Duan Aixia, Chen Jing, Liu Hongde, et al. Application and development of molecular docking method[J]. *Journal of Analytical Science*, 2009, 25(4): 473-477.
- [5] Moont G, Gabb H A, Sternberg M J E. Use of pair potentials across protein interfaces in screening predicted docked complexes[J]. *Proteins*, 1999, 35(3): 364-373.
- [6] Sanders J, Kandrot E. *CUDA by example: An introduction to general-purpose GPU programming*[M]. Upper Saddle River, New Jersey, USA: Addison-Wesley, 2011.
- [7] Storti D, Yurtoglu M. *CUDA for engineers: An introduction to high-performance parallel computing*[M]. Upper Saddle River, New Jersey, USA: Addison-Wesley, 2017.
- [8] Shatsky H J. Flexible protein alignment and hinge detection [J]. *Proteins*, 2002, 48(2): 242-256.
- [9] Bonvin A M. Flexible protein-protein docking[J]. *Current Opinion in Structural Biology*, 2006, 16 (2): 194-200.
- [10] 赵剑, 阮越, 王嘉松. 数学结构的蛋白质二维数字表达及其应用[J]. *数据采集与处理*, 2013, 28(6): 770-776.  
Zhao Jian, Ruan Yue, Wang Jiasong. 2-D numerical representation of protein based on mathematical structure and its application[J]. *Journal of Data Acquisition and Processing*, 2013, 28(6): 770-776.
- [11] 龚新奇, 刘斌, 常珊, 等. 蛋白质复合物结构预测的集成分子对接方法[J]. *中国科学(C辑): 生命科学*, 2009, 39(10): 963-973.  
Gong Xinqi, Liu Bin, Chang Shan, et al. Integrated molecular docking method for prediction of protein complex structure [J]. *Chinese Science (Series C): Life Sciences*, 2009, 39(10): 963-973.

- [12] 吴奎, 宋彦, 戴礼荣. 基于 CUDA 的 GMM 模型快速训练方法[J]. 数据采集与处理, 2012, 27(1):85-90.  
Wu Kui, Song Yan, Dai Lirong. CUDA based fast GMM model training method and its application[J]. Journal of Data Acquisition and Processing, 2012, 27(1):85-90.
- [13] 李正夫, 王希诚, 郭权. CUDA 下受体评分网格生成并行算法[J]. 计算机应用研究, 2013, 30(3): 814-816.  
Li Zhengfu, Wang Xicheng, Guo Quan. Parallel algorithm for grid generation of receptor score under CUDA[J]. Computer Application Research, 2013, 30(3):814-816.
- [14] Vakser I A. Protein-protein docking: From interaction to interactome[J]. Biophysical Journal, 2014, 107(8): 1785-1793.
- [15] Xu Y, Shen J, Luo X, et al. How does huperzine A enter and leave the binding gorge of acetylcholinesterase? Steered molecular dynamics simulations[J]. Journal of the American Chemical Society, 2003, 125 (37): 11340-11349.
- [16] Lensink M F, Wodak S J. Docking, scoring, and affinity prediction in CAPRI[J]. Proteins: Structure, Function, and Bioinformatics, 2013, 81(12): 2082-2095.
- [17] Janin J. Protein-protein docking tested in blind predictions: The CAPRI experiment[J]. Molecular Biosystems, 2010, 6 (12): 2351-2362.
- [18] Ma X H, Li C H, Shen L Z, et al. Biologically enhanced sampling geometric docking and backbone flexibility treatment with multiconformational superposition[J]. Proteins-Structure Function and Bioinformatics, 2005, 60(2): 319-323.
- [19] Lam P Y, Jadhav P K, Eyermann C J, et al. Rational design of potent, bioavailable, nonpeptide cyclic ureas as HIV protease inhibitors[J]. Science, 1994, 263(5145): 380-384.
- [20] Tao P, Lai L. Protein ligand docking based on empirical method for binding affinity estimation[J]. Journal of Computer-Aided Molecular Design, 2001, 15(5): 429-446.
- [21] Mohan V, Gibbs A C, Cummings M D, et al. Docking: Successes and challenges[J]. Current Pharmaceutical Design, 2005, 11(3): 323-333.

## 作者简介:



陆旭峰(1993-),男,硕士研究生,研究方向:GPU 并行计算方法, E-mail: xflu0913@163.com。



陆振宇(1994-),男,硕士研究生,研究方向:GPU 并行计算方法。



梅向东(1968-),男,高级工程师,研究方向:图形图像处理技术。



孔韧(1981-),女,博士,研究员,研究方向:计算机辅助药物设计方法等。



常珊(1982-),男,通信作者,教授,研究方向:生物信息学和并行计算等, E-mail: schang@jsut.edu.cn。

(编辑:张黄群)