中国科学: 信息科学 2012 年 第 42 卷 第 8 期: 1034–1048
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS
论 文

从头预测蛋白质骨架的一种并行蚁群方法及其在 CASP8/9 中的应用

吴宏杰[①③], 吕强[①②*], 吴进珍[①], 黄旭[①], 罗小虎[①], 钱培德[①②]

① 苏州大学计算机科学与技术学院, 苏州 215006
② 江苏省计算机信息处理技术重点实验室, 苏州 215006
③ 苏州科技学院电子与信息工程学院, 苏州 215009
* 通信作者. E-mail: qiang@suda.edu.cn

收稿日期: 2011–03–15; 接受日期: 2011–07–18
国家自然科学基金 (批准号: 60970055) 资助项目

摘要 从低同源关系的氨基酸序列预测蛋白质的三维结构被称为从头预测, 它是计算生物学领域中的挑战之一. 蛋白质骨架预测是从头预测的必要先导步骤. 本文应用一种基于共享信息素的并行蚁群优化算法, 在现有能量函数指导下, 通过不同能量项之间的定性互补, 构建具有最低能量的蛋白质骨架结构, 并通过聚类选择构象候选集合中具有最低自由能的构象. 在 CASP8/9 所公布的从头建模目标上应用了该方法, CASP8 的 13 个从头建模目标中, 模型 1 中有 2 个目标的预测结果超过 CASP8 中最好的结果, 7 个位列前 10 名; CASP9 的 29 个从头建模目标中, 候选集中的最佳结果中有 20 个进入 Server 组的前 10 名, 模型 1 中有 11 个进入前 10 名. 本文的结果说明融合多个不同的能量函数指导并行搜索, 可以更好地模拟天然蛋白质的折叠行为. 同时, 在本算法载体上实现了不同种类搜索策略的融合并行, 对于用非确定性算法解决类似的优化问题来说也是一种新颖的方法.

关键词 蛋白质骨架 从头预测 蛋白质折叠 并行算法 启发式算法

1 引言

蛋白质的三维结构决定了其生化功能, 而采用生化手段测定蛋白质结构代价高、耗时长. 因此, 采用计算的手段测定蛋白质三维结构成为计算生物学中的重要课题之一. 目前的蛋白质结构预测方法根据其序列的同源性可分为两大类 [1,2]: 基于模板的建模 (templete based modeling) 与自由建模 (free modeling, FM). FM 所针对的目标序列同源性低于 30%, 没有足够精确的三维结构信息, 因此又称为从头 (*de novo*) 预测. 在从头预测中, 确定蛋白质骨架是蛋白质全原子结构预测的先导步骤. 同时, 有理论认为, 骨架对蛋白质整个折叠过程和生化性能的决定有着很重要的作用 [3]. 所以, 蛋白质骨架的预测具有工程和科学意义. 所谓从头预测蛋白质骨架是指仅根据蛋白质的一维氨基酸序列, 在没有合适模板的情况下, 基于最小自由能原理 [4] 预测出描述蛋白质骨架的二面角序列, 为后续全原子预测提供粗粒度的空间构象.

引用格式: 吴宏杰, 吕强, 吴进珍, 等. 从头预测蛋白质骨架的一种并行蚁群方法及其在 CASP8/9 中的应用. 中国科学: 信息科学, 2012, 42: 1034–1048

在两年一度的蛋白质结构预测评估大会 CASP (critical assessment of techniques for protein structure prediction)[5] 中, 从头预测最具有挑战性. CASP 在世界范围内对蛋白质结构预测技术进行评估, 评价当前蛋白质结构建模方法的能力与局限, 被称为蛋白质结构预测领域的奥运会. 最近的 CASP9 在 2010 年 8 月完成. 在历届 CASP 比赛中, David Baker 小组的 Rosetta 和 Zhang Yang 小组的 I-TASSER/Zhang-Server 均有突出的表现. 前者在 CASP1∼6 中占绝对优势, 后者在 CASP7∼9 中始终位于前三名. Rosetta 采用 Monte Carlo 与模拟退火方法, 通过片段组装的方法构造完整的蛋白质骨架, 然后通过全原子细化得到更高精度的蛋白质构象 [6,7]. 而 I-TASSER 首先采用串线方法 (threading) 找出连续的蛋白质局部结构, 然后通过局部结构组装成蛋白质构象, 最后利用多次 Monte Carlo 对聚类中心进行细化 [8,9].

随着计算机单机性能的飞速提高和并行计算构架的流行, 采用并行计算来预测蛋白质骨架成为一种自然的选择. 文献 [10] 实现了把 Rosetta 的预测协议中显式的独立操作用 OpenMP 分段并行的方案, 并测试了 4 个蛋白质的预测结构. 文献 [11] 展示了用一个松散的计算网格来预测蛋白质结构的方案, 并用了 2 个测试用例演示了并行计算的强大. 文献 [12] 用并行产生每一候选结构的方法, 在 65536 个 CPU 上用半天时间实践了一个 CASP7 预测目标. 上述这些方案的特点是把并行计算平台作为一个超级快的 CPU 来使用, 本质上并没有改变现有预测协议的逻辑, 从而预测精度上也没有本质提高.

一般来说, 蛋白质结构从头预测有两个关键难点 [2,7]: 一、由于有机分子及其内部微粒之间关系的复杂性, 目前的能量函数都不够准确; 二、蛋白质结构中存在大量的自由度, 导致构象的搜索空间巨大. 针对第一个难点, 本文的措施是把蛋白质骨架预测问题模型化为多目标优化问题. 文献 [13, 14] 尝试用了最简单的两种解决多目标优化的方法, 把 3 个简单的能量项作为 3 个最小化的目标函数, 对 CASP8 的 4 个案例进行了测试. 该方案只是考虑融合了不同的能量项, 并没有考虑把多种搜索算法的特色发挥出来.

本文认为, 能量函数是解决蛋白质骨架预测问题中领域知识的高度浓缩, 例如 Rosetta 的能量函数, 被实践证明优质的能量函数, 应该得到充分地利用. 其次, 各种构象搜索算法针对不同的能量地形有不同的性能, 在并行计算的平台上, 并行融合多种搜索算法的智慧是可行的. 因此, 本文把结构预测中的并行融合多种搜索方法作为研究重点, 能量函数直接采用 Rosetta 中的多个能量函数. 本文将 Rosetta 的多个能量函数用并行计算方法进行融合, 以 ACO (ant colony optimization) 算法 [15,16] 作为基本搜索载体, 设计和实现了并行 ACO 算法 pacBackbone (parallel ant colonies for backbone prediction) 来预测蛋白质骨架, 让不同的能量函数同时发挥作用, 使启发式搜索能够有效地发现低能量结构. 该方法在 CASP8/9 共 42 个 FM 案例上进行测试, 取得了满意的结果.

2 问题描述

到目前为止, 所有的蛋白质结构从头预测方法都以 Anfinsen 假说 [4] 为依据, 把预测问题建模成一个优化问题. 所以, 从头预测骨架的面向计算的问题描述就从搜索空间和能量函数两个方面开始.

本文从头预测的搜索空间产生于已知结构的蛋白质骨架片段. 基于片段的从头预测方法将已知结构的蛋白质片段组装出大量的预测目标三维结构, 再依据某种评价标准从中挑选出近天然的结构 [6,17]. 预测过程中用到的片段信息来于 Robetta 在线预测服务器 [18], 针对每个预测目标生成了 3 残基与 9 残基两个片段库 $\mathcal{F}_3$ 和 $\mathcal{F}_9$. Robetta 片段库所使用的是非冗余的结构数据库, 在生成片段库过程中过滤掉了预测目标的同源蛋白质结构的相关数据. 从这个意义上来说, 基于这样的片段库的预

测, 可以认为是从头预测, 特别针对 CASP8/9 中的 FM 预测目标是很适合的.

一般把氨基酸序列的二级结构分为 3 大类, 第 1 类为螺旋 (helix); 第 2 类为折叠 (sheet); 除螺旋与折叠之外的其他二级结构统称为环区 (loop). 在本文中分别用 H, E 和 L 来表示. 蛋白质骨架结构采用一组二面角序列表达. 可以认为每个残基对应于一个四元组 (ϕ,ψ,ω,ss), 其中 ϕ,ψ,ω 分别描述骨架的 3 个二面角, ss 表示残基的二级结构标签. 因此, 蛋白质骨架预测以氨基酸序列作为输入, 以四元组序列作为输出. 实际上, 片段的引入把原来的连续搜索空间转换为离散的搜索空间, 而这种离散空间是基于已知蛋白质结构信息. 除了上述 $\mathcal{F}_3$ 和 $\mathcal{F}_9$ 外, 针对环区重构阶段, 本文还生成了 1 残基片段库 $\mathcal{F}_1$. 所谓 1 残基片段库, 就是将 3 残基和 9 残基的片段拆分合并成片段长度为 1 的片段库. 不管哪一种类型的片段库, 每个片段还包含该片段与预测目标链的相似度分值. 所以最终每个片段可以用一个五元组列表表示 $f=\{(\phi,\psi,\omega,ss,\eta)_1,\ldots,(\phi,\psi,\omega,ss,\eta)_s\}$, 其中 η 为相似度分值, 在后文中将此作为算法的启发值; s 表示片段长度, 在本文中为 1, 3 和 9. 对于骨架上的第 i 号残基, 有一组片段构成其搜索空间, 用 F_i 表示这样的片段集合. 对于一个长度为 n 残基序列, 就有由 n 个 F_i 构成片段库 $\mathcal{F}$.

片段组装方式得到的骨架, 在不同的优化阶段用不同的能量函数进行评价, 从而探索质量最好的那些构象. 能量函数一般主要考虑以下因素: 空间的排斥力、环境作用、残基对作用、二级结构之间组装关系、链的紧密程度、排除溶剂的体积等等 [6]. Rosetta 3.x 系统 [19] 的从头预测蛋白质骨架程序中使用了 5 个能量函数 score0, score1, score2, score3, score5, 它们都包含了上述的能量项, 但能量项的权重不同. Rosetta 的从头预测算法先顺序使用 score0 与 score1, 然后交替使用 score2 与 score5, 最后用 score3 对骨架进行优化. 本文将这 5 个能量函数作为组合优化问题的多个目标函数. 值得说明的是, 从计算的角度来说, 本文的方法独立于能量函数的选择.

至此, 蛋白质骨架预测问题已经被映射为面向计算的多目标组合优化问题.

3 并行蚁群解决方案

3.1 单蚁群的设计

本文基于给定的能量函数, 采用 ACO 算法在片段空间中搜索最低能量值的骨架. 单蚁群算法描述如算法 1 所示.

算法 1 第 2 行 Initialize() 对算法参数进行初始化, 根据蛋白质序列 $\mathcal{S}$ 初始化蛋白质构象 $\mathcal{M}$, 将所有位置的残基三个二面角度分别设置为 $-150°$, $150°$ 和 $180°$, 二级结构标签 ss 设置为 L. 初始化全局最优解 $\mathcal{M}_{bs}$ 为 $\mathcal{M}$.

算法 1 第 3 行通过算法控制参数 p_ac 调节蚁群迭代的次数. 第 4 行的循环为蚁群中每一个蚂蚁构造骨架, p_ant 用于控制蚁群中蚂蚁的个数, 同样也是一个算法控制参数. 第 5 行的循环为单个蚂蚁构造最优骨架而进行的片段插入, p_cc 用于控制单个蚂蚁片段插入的次数.

算法 1 第 6 行从 3 残基片段库 $\mathcal{F}_3$ 或 9 残基片段库 $\mathcal{F}_9$ 中选择一种作为当前片段库 $\mathcal{F}$, 选择的依据为

$$\mathcal{F}=\begin{cases}\mathcal{F}_3, & \text{if } q\leqslant q0,\\ \mathcal{F}_9, & \text{otherwise},\end{cases} \tag{1}$$

其中 q 为 [0,1] 之间产生的一个随机数, $q0$ 为一个常数, 用于调节选择 $\mathcal{F}_3$ 与 $\mathcal{F}_9$ 的概率.

算法 1 第 7 行为当前片段随机确定一个插入位置. 虽然一个 n 长度的蛋白质最少只需要 $[n/9]$ 次

算法 1 单蚁群算法 AC($\mathcal{S}$, n, $\mathcal{F}_3$, $\mathcal{F}_9$, $\mathcal{T}$, E)

1: Input: 蛋白质 fasta 序列 $\mathcal{S}$, 蛋白质长度 n, 3 残基片段库 $\mathcal{F}_3$, 9 残基片段库 $\mathcal{F}_9$, 信息素矩阵 $\mathcal{T}$, 能量函数 E
2: Initialize();
3: **for** $gen = 1$ **to** $n * p\_ac$ **do**
4: **for** $j = 1$ **to** $n * p\_ant$ **do**
5: **for** $it = 1$ **to** $n * p\_cc$ **do**
6: 根据公式 (1), 选择片段库 $\mathcal{F}_3$ 或 $\mathcal{F}_9$, 设为 $\mathcal{F}$;
7: 随机选择一个残基位置 i, 从 $\mathcal{F}$ 中, 确定为第 i 组片段集合 F_i;
8: 根据公式 (2) 选择片段 f_j^*;
9: $\mathcal{M}_j \leftarrow$ SubstituteF(f_j^*, $\mathcal{M}_j$);
10: **end for**
11: **end for**
12: $\mathcal{M}_{ib} \leftarrow \text{argmin}_E(E(\mathcal{M}_1), \ldots, E(\mathcal{M}_{n*p\_ant}))$;
13: $\mathcal{M}_{ib} \leftarrow$ LocalOptimization($\mathcal{M}_{ib}$);
14: UpdatePheromone($\mathcal{M}_{ib}$);
15: **if** $E(\mathcal{M}_{ib}) \leqslant E(\mathcal{M}_{bs})$ **then**
16: $\mathcal{M}_{bs} \leftarrow \mathcal{M}_{ib}$;
17: **then**
18: **end for**
19: $\mathcal{M}_{bs} \leftarrow$ LoopRebuild($\mathcal{M}_{bs}$).
20: Output: 最优构象 $\mathcal{M}_{bs}$

9 片段插入就可以完成一次所有残基的替换. 但是由于候选片段序列与目标序列具有同源相似性, 而非绝对相等, 所以我们这里尝试片段覆盖插入法, 即随机确定插入位置, 且允许覆盖.

算法 1 第 8 行从当前片段库 $\mathcal{F}$ 中选择一个最优片段 f_j^*, 选择的依据为

$$f_j^* = \begin{cases} \text{argmax}_{f_j \in F_i}[\tau_{ij}]^\alpha [\eta_{ij}]^\beta, & \text{if } q \leqslant q1, \\ \text{random pick up a } f_j \text{ from } F_i, & \text{otherwise}, \end{cases} \tag{2}$$

其中 F_i 是候选片段的集合; τ_{ij} 是信息素, η_{ij} 是启发值, 它们的下标 i, j 分别对应第 i 号残基可选的在 F_i 中的片段序号 j; 参数 α 和 β 是用来调节蚁群算法中的信息素和启发值的权重, 即调节蚁群依赖历史搜索经验与当前所见启发之间的比重; q 为 [0,1] 之间产生的随机数, 参数 $q1$ 可以调节按照随机方式或按照概率模型选择片段的比重.

算法 1 第 9 行 SubstituteF(f_j^*,$\mathcal{M}_j$) 将片段 f_j^* 中的二面角和二级结构标签替换构象 $\mathcal{M}_j$ 对应残基处的角度和标签. 第 12 行表示从蚁群中选择一个能量最低构象作为本代的最优构象 $\mathcal{M}_{ib}$.

算法 1 第 13 行 LocalOptimization($\mathcal{M}_{ib}$) 是在单个蚂蚁完成骨架构造后进行的局部优化, 采用贪婪和模拟退火相结合的方法, 对于随机选择的一个片段位置, 在其片段集合中挑选一个能够使当前能量 “降低” 的片段来插入. “降低” 是按照 Metropolis[20] 规则来判定.

算法 1 第 14 行 UpdatePheromone() 采用基于 MMAS[21] 的全局信息素更新规则: 选择迭代最好解进行信息素更新. 新的信息素 τ'_{ij} 计算公式如下:

$$\tau'_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}, \tag{3}$$

其中参数 $\rho \in [0,1)$ 为信息素挥发系数; $\Delta\tau_{ij} = Q(E(\mathcal{M}))$. 由于能量值值域过于宽泛, 所以把能量值 $E(\mathcal{M})$ 用质量函数 Q 映射到一个特定的区间, 本文中 Q 采用反余切函数.

算法 1 第 15~17 行将更优的本代最优构象 $\mathcal{M}_{ib}$ 替换全局最优构象 $\mathcal{M}_{bs}$.

最后, 算法 1 第 19 行 LoopRebuild($\mathcal{M}_{bs}$) 针对 $\mathcal{M}_{bs}$ 的 loop 区进行局部优化. 由于 L 结构较 H 和 E 具有更大的自由度, 因此在蛋白质结构预测中更为困难. 本文根据已有的 $\mathcal{F}_3$ 和 $\mathcal{F}_9$ 生成新的粒度更细的 $\mathcal{F}_1$ 残基片段库. 采用 Monte Carlo 搜索结合 Metropolis 规则, 在 $\mathcal{F}_1$ 中搜索二级结构为 L 的 1 残基片段, 试图降低全局骨架的能量值.

3.2 并行蚁群设计与实现

共享信息素矩阵策略是一种非常有效的并行 ACO 策略 [22], 并行的多个蚁群共享一个信息素矩阵, 每个蚁群可以有不同的搜索策略和目标函数. 通过这个共享信息素矩阵, 每个蚁群可以同步地交换搜索经验, 共同朝着更好的目标值演化. 共享信息矩阵策略已经成功地应用到诸如 TSP[22]、Bayes 网络学习 [23]、蛋白质 HP 折叠问题 [24] 和 QAP[25] 上.

利用 OpenMP 使这种并行方法实现更加简单, 每个蚁群独立运行在各自的线程中, 除信息素矩阵外其他数据都是私有的. 为了能使解更富有多样性, 本文对于每个蚁群对信息素矩阵的访问不加并发控制, 希望这种不同步的更新和读取信息素可以更合理地将多个蚁群的搜索经验融合.

具体地, pacBackbone 派生 $p > 4$ 个并行的蚁群线程 AC_1(score0), ..., AC_p(score3). 前 4 个线程的蚁群分别用 score0, score1, score2, score5 指导搜索最优解. 其余的蚁群以 score3 为指导. pacBackbone 的优势在于并行的蚁群共享一个信息素矩阵 $\mathcal{T}$. 蚁群中的优质搜索轨迹被间接记录在同一个矩阵 $\mathcal{T}$ 中. 因此, 不仅仅是蚁群中蚂蚁之间在同一能量函数的指导下协同, 而且各蚁群在不同的能量函数指导下朝着优质结构方向协同进化.

3.3 聚类 decoys

由于现有的能量函数不够精确, 所以, 最低能量的骨架未必最接近天然结构. 通常的处理方法是产生多个预测结构构成集合, 这个集合被称为 decoys. 通过对 decoys 中结构空间相似性为依据的聚类, 来找出最具代表性的骨架, 以此作为预测的最终结构. CASP 要求预测者对每个预测目标提交 5 个结构, 并将这 5 个结构按预测者的准则从优到劣排序, 编号为 1 的结构 (称为模型 1) 也就是预测者提交的 “最优” 结构, 这是 CASP 最主要的评估对象.

Rosetta3.x 中的聚类协议 [19] 基于结构 RMSD (root mean square deviation) 相似度, I-TASSER 的聚类协议 SPICKER [26] 采用 TM-score (template modeling score) 进行聚类. 由于聚类方法各有优缺点, 结构相似性尺度的选择也会与能量函数表现自由能的程度有关 [27], 所以针对不同的预测目标, 本文采用了两种不同的聚类方法. 对于预测目标结构域连续的采用了基于 RMSD 相似度的 Rosetta3.x 聚类协议; 而对于不连续的则采用基于加权 GDT_TS (global distance test total score) 的 AP (affinity propagation) 聚类方法 [28].

AP 聚类算法是一个比较新颖的聚类算法, 对聚类目标的相似度处理比较灵活. 由于本文采用的能量函数比较偏好残基之间的局部作用 [6], 所以, 对于不连续的结构域的预测目标来说 (例如 CASP8 目标 T482-D1), 能量函数反映的误差要大一些. 我们认为在这种情况下, 当两个构象的能量差异太大时, 即使它们的结构很相似, 也不倾向把它们聚在一个类中. 为此, 我们需要对构象之间的相似度根据能量差异进行加权处理. 首先, 相似度用 GDT_TS 来衡量, 假定两个构象 i 和 j 之间的相似度是 ξ_{ij},

我们同时计算 i 和 j 之间的能量差 ΔE_{ij}, 以

$$w_{ij} = \frac{\Delta E_{ij}}{\sum_{p,q\in \text{decoys}} \Delta E_{pq}}$$

为基础权值, 其值越大, 对 ξ_{ij} 的惩罚越大. 经过归一化处理, 修正后的权值是 $w'_{ij} = \frac{2}{N} - w_{ij}$, 其中 N 是 decoys 中所有配对的总个数. 传递给 AP 聚类算法的相似度是 $\xi'_{ij} = \xi_{ij}(1 + w'_{ij})$.

4 对 CASP8/9FM 目标的评测

4.1 测试平台和测试集

本文中 CASP8 目标预测所用的硬件环境为 4 路双核 64 位 1.6 GHz Power (gr) CPU 的 IBM pServer, 所以我们设置并行线程个数 $p = 8$. CASP9 目标预测所用的硬件环境为曙光高性能集群, 包含 20 个 4 路 AMD8347 低功耗四核处理器的计算节点, 共计 320 个计算单元, 所以我们设置并行线程个数 $p = 16$.

本文并行计算中的评分函数采用了 Rosetta 提供的能量函数, 其中有 4 个线程分别采用 score0, score1, score2, score5, 其余线程全部采用 score3; 在局部优化阶段, 所有线程都采用 score3. 聚类过程中, 需要设置聚类的参数, 其中 Rosetta3.x 的聚类协议中聚类半径 radius 设置为 3.0 (对 CASP8) 和 7.0 (对 CASP9), AP 加权聚类中的聚类半径设为默认的 median. CASP8 中的每个目标我们生成了 800 个 decoys, CASP9 中的每个目标我们生成了 1600 个 decoys. 根据经验, 算法 1 中 3 个用于控制循环次数的参数 p_ac, p_ant 和 p_cc 分别设为 0.6, 0.6 和 6; 式 (1)∼(3) 中的参数 α, β, $q0$, $q1$ 和 ρ 也分别设为 1.0, 1.0 , 0.6, 0.8 和 0.01.

CASP8 有 13 个 FM 目标, CASP9 有 29 个 FM 目标, 详细标号请参见 CASP 网站 [5] 或本文在线补充材料表 S1[1)]. 本文以这两组预测目标为测试集.

对于从头预测骨架质量评估标准, CASP 官方采用的指标有 GDT_TS 和 RMSD 等. GDT_TS 越大表示和天然结构越相似, RMSD 越小表示与天然结构越相似. 为了考察预测方法在不同评价指标下的预测精度, 本文将 GDT_TS, Q_{short} 和 Q_{long}[29] 作为 CASP8 预测目标的评测指标, 将 RMSD 作为 CASP9 预测目标的评测指标. 事实上, 骨架质量评估本身是一个非常困难的问题, 参见 5.2 小节的详细讨论. 所有的比较, 我们从纵横两个层面展开. 在纵向, 对于每一个 FM 目标, 我们都给出了 pacBackbone 绝对的精度及其相对所有 CASP 参评系统的排名. 在横向方面, 我们给出所有 FM 目标评测结构的综合排名.

CASP 比赛中各参赛队可以分为 Server 和 Human 两组, 前者是仅由机器自动进行预测, 整个预测过程中没有人类专家的交互帮助; 后者是机器预测过程中可以引入领域专家的人工干涉与指导. 本文的方法基于全自动的计算方法, 没有引入额外的专家干涉, 所以本文的性能评价在 Server 组中进行, CASP8 有 122 个 Server 组, CASP9 有 139 个 Server 组 [5]. 本文的结果根据 CASP 官方公布的结果进行了比较.

4.2 对 CASP8 FM 测试集的结果比较

我们分析了对 CASP8 的 13 个 FM 目标的预测结果. 详细数据表格参见本文在线补充材料表 S2

1) 本文可运行的程序、完整的数据集、表格结果和高分辨插图, 可以从 http://ckcst20.suda.edu.cn:8080/pacBackbone2 获得

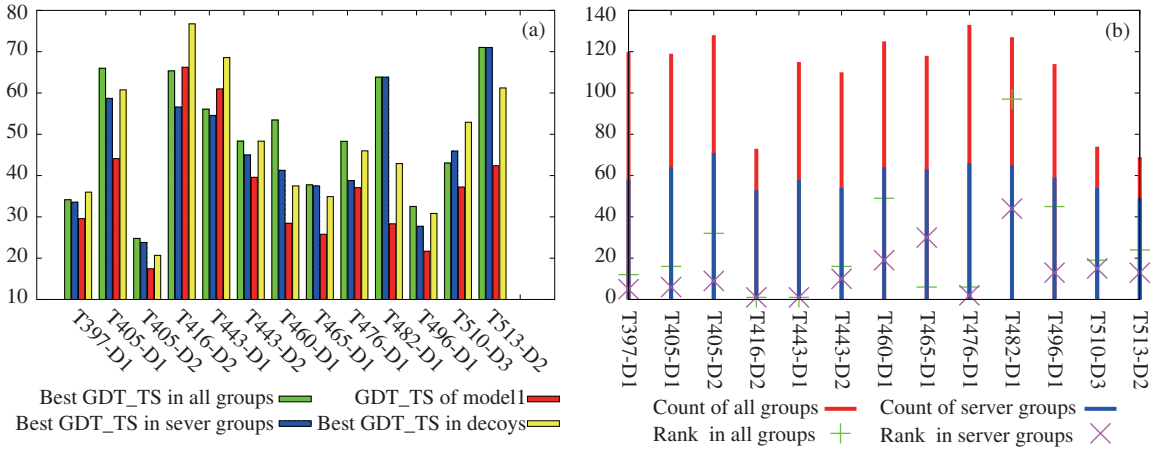

图 1　pacBackbone 与 CASP8 所有提交结果的比较情况

Figure 1　The comparisons between pacBackbone and all other submitted results of CASP8. (a) GDT TS of CASP8 FM target; (b) rank of pacBackbone model 1

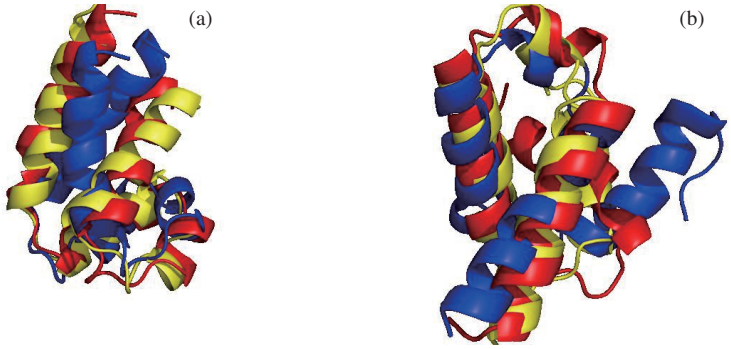

图 2　目标 T416-D2(a) 和目标 T443-D1(b) 预测骨架的叠加比较情况: 天然结构 (红色), CASP8 中 Server 类最好结构 (蓝色) 和本文预测结构 (黄色)

Figure 2　The superposition comparison with the native structures (red), predicted model1 (yellow) and the best predicted model1 of the CASP8 Server group (blue) for proteins T416-D2 (a) and T443-D1 (b)

(见第 1039 页脚注 1)). 图 1 给出了对每个目标的绝对精度比较, 以及排名比较. 图 1(a) 中绿色与蓝色柱体分别表示所有参赛组与 Server 组预测结果的 GDT_TS 的最优 (最大) 值. 红色柱体表示 pacBackbone 所预测的模型 1 的 GDT_TS 分数, 其中有 2 个目标 (T416-D2, T443-D1) 的 GDT_TS 分数超过了所有参赛组的最优值. 黄色柱体表示 pacBackbone 产生的 decoys 中的最优 GDT_TS 分数, 其中有 5 个目标在所有参赛组中最优值, 有 8 个目标在 Server 组中最优. 红色柱体与黄色柱体的比较说明, pacBackbone 获得了高质量 decoys, 但由于聚类方法对近天然结构的分辨能力还不够强, 使得聚类后的模型 1 分数较聚类前降低了不少. 图 1(b) 中比较了 pacBackbone 模型 1 的 GDT_TS 分数在所有参赛组与 Server 组中的排名情况, 红色柱体高度为所有小组的个数, 绿色标记为 pacBackbone 在所有小组中的排名; 蓝色为 Server 小组的个数, 紫色标记为 pacBackbone 在 Server 小组中的排名. pacBackbone 有 7 个目标在 Server 组中位列前 10 名, 所有目标中只有一个目标 T482-D1 排在 Server 组的半数之后, 其余 12 个目标全部排在半数之前.

图 2 所示为 pacBackbone 排名第一的目标 T416-D2 与 T443-D1 叠加天然结构的比较细节 (由

PyMOL[30] 生成).

另外, 更加细节的 CASP8 的 13 个 FM 目标的 GDT_TS 细节分析图参见本文在线补充材料 S2 (见第 1039 页脚注 1)).

GDT_TS 是一种基于序列对齐的三维结构比较方法, 虽然在自由建模中使用最为广泛, 但该类方法存在局部结构的较小偏差容易引起评价分数的较大偏差的问题. Q 分数 [29] 是基于两个结构的内部距离的比较方法, 内部距离计算了单个模型中某残基 Ca 与其他所有残基 Ca 之间的距离, 很好地避免了序列对齐的依赖性. 所以本文除了使用 GDT_TS 进行比较外, 还使用 Q_{short} (残基对相隔小于 20 的 Q 分数) 与 Q_{long} (残基对相隔大于 20 的 Q 分数) 分数进行单个目标的分析. pacBackbone 相对于上述两系统仍具有一定的优势, 详细分析图参见本文在线补充材料 S3 (见第 1039 页脚注 1)).

最后, 我们根据 z-score 来考察 pacBackbone 在 CASP8 的所有 13 个 FM 目标预测的综合性能. CASP8 的最终综合排名是以 Z-M1-GDT_TS 高低排序 (越大越好), 也就是累加每个预测目标模型 1 的 z-score. 本文将这 13 个 FM 预测目标与 ZhangServer 和 Rosetta 两个小组比较, 结果如图 3 所示, 详细数据表格参见本文在线补充材料表 S3 (见第 1039 页脚注 1)).

图 3 中绿色与蓝色柱体分别为 ZhangServer 与 Rosetta 的 Z-M1-GDT_TS 分数, 红色柱体为 pacBackbone 模型 1 的 Z-M1-GDT_TS 分数, 其中有 7 个目标同时优于这两个对比小组. 从右侧图中的综合分数 Z-M1-GDT_TS 来看, pacBackbone 的性能以 0.03 的微弱差距次于 Rosetta, 以 2.45 的较大优势领先于 ZhangServer. 值得说明的是, 这两个小组在 CASP8 的 Server 组中排名为第一和第三名.

4.3 对 CASP9 FM 测试集的结果比较

CASP9 于 2010 年 5 月 24 日发布了第一个预测目标, 每个工作日发布 3 个目标, 每个目标可以有 72 小时的计算时间. pacBackbone 以 LenServer 为名参加了 Server 组的比赛, 对所有 118 个预测目标共提交 588 个结果. 我们对于每个 CASP9 FM 目标大约使用了小于一千小时 CPU 的计算资源, 平均每个目标生成 717 个候选结构. 这些资源相对于 ROBETTA 的 Blue Gene 超级计算机, 以及几十万小时的计算机代价 [12,31] 是相当有限的. 尽管如此, 在 CASP9 官方公布的比赛结果中 LenServer 在所有 78 个进行 FM 预测的 Server 组中综合排名 26[5], 且有两个 FM 目标 T0555-D1 和 T0604-D3, 分别取得了第二和第三名. 这个两个目标的 GDT 分析如图 4 所示.

图 4 中红色线条 pacBackbone 的结果, 同 GDT 阈值下 CA 所占的比例明显高于 ZhangServer (蓝色) 与 Rosetta (绿色) 的结果.

为了能在尽量相当的计算资源情况下, 测试 pacBackbone 算法预测精度, CASP9 赛后我们对除了 T0555-D1 与 T0604-D3 之外的 27 个 FM 目标进行了重做, 延长了每个目标的计算时间, 使每个目标的 decoys 数目达到 1600. 对 CASP9 FM 目标延长计算时间后的 RMSD 与排名情况图 5 所示, 详细数据表格参见本文在线补充材料表 S4 (见第 1039 页脚注 1)).

图 5(a) 是 CASP9 各个 FM 目标 pacBackbone 与 Server 组最优 (最小) RMSD 比较, 蓝色柱体是 Server 组中 RMSD 的最优值, 红色与黄色柱体分别表示 pacBackbone 模型 1 与 decoys 中的最优 RMSD. 图 5(b) 是 pacBackbone 在 Server 组的排名情况, 红色柱体是 Server 组个数, 绿色标记与蓝色标记分别是 pacBackbone 模型 1 与 decoys 中的最优模型的排名. pacBackbone 所预测的 decoys 中最优结果有 20 个进入了 Server 组的前 10 名. 经聚类后挑选出的模型 1 中有 11 个目标 (包括 T0555-D1 与 T0604-D3) 进入前 10 名.

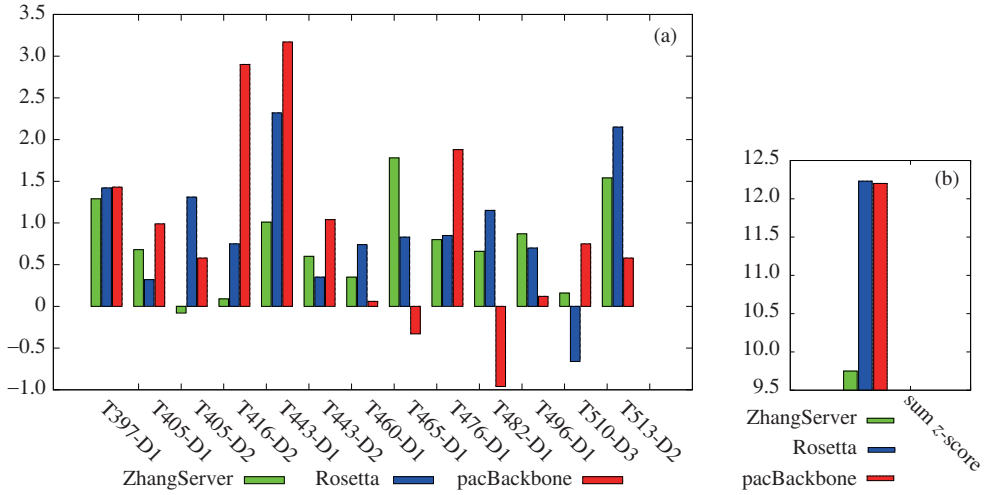

图 3　pacBackbone 与 CASP8 领先系统在所有 FM 目标预测综合 Z-M1-GDT_TS 比较

Figure 3　The Z-M1-GDT_TS comparison between pacBackbone and the leaders of CASP8. (a) CASP8 FM target z-score(GDT TS); (b) sum of z-score

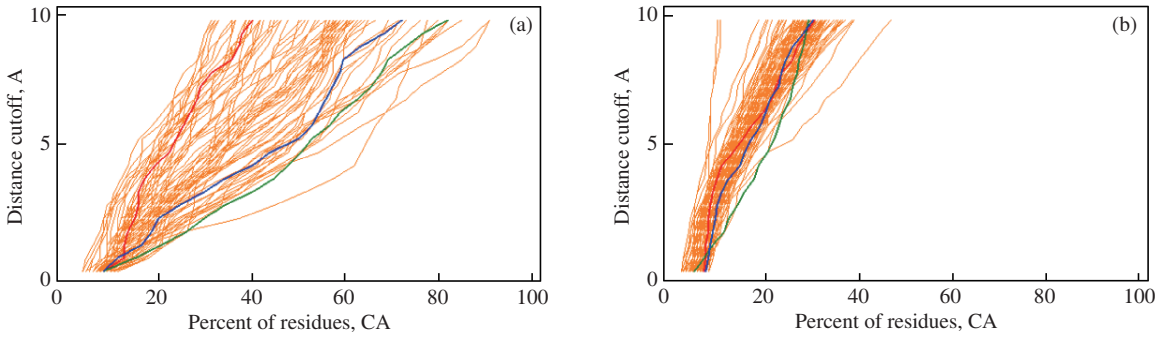

图 4　T0555-D1 与 T0604-D3 的预测精度细节分析图

Figure 4　The GDT_TS plots of T0555-D1 and T0604-D3 predictions for all submitted model1s. (a) T0555-D1; (b) T0604-D3

最后, 我们根据 z-score-RMSD (越小越好) 来考察 pacBackbone 与其他 CASP9 优秀参赛系统的综合性能. 图 6 是 pacBackbone 在 Server 组范围内, 与 ZhangServer (绿色)、Rosseta (蓝色)、QUARK (紫色, CASP9 官方公布的 Server 组第一名) 的 Z-M1-RMSD 的比较结果, 详细数据表格参见本文在线补充材料表 S5 (见第 1039 页脚注 1)). 红色与黄色柱体分别是 pacBackbone 产生的模型 1 与 decoys 中最优结果的 z-score-RMSD. 在单个目标的 z-best-RMSD 比较中, pacBackbone 的 decoys 最优结果有 2/3 的目标同时优于其他 3 组的结果. 模型 1 的 Z-M1-RMSD 有 10 个目标同时优于其他 3 组的结果, 有 2/3 的目标至少优于其中一组. 右侧图显示 pacBackbone 的模型 1 的综合分数 Z-M1-RMSD 优于 Rosetta.

5　讨论

本文报告了一种基于并行蚁群的蛋白质三维结构预测方法 pacBackbone. 采用并行算法解决从头

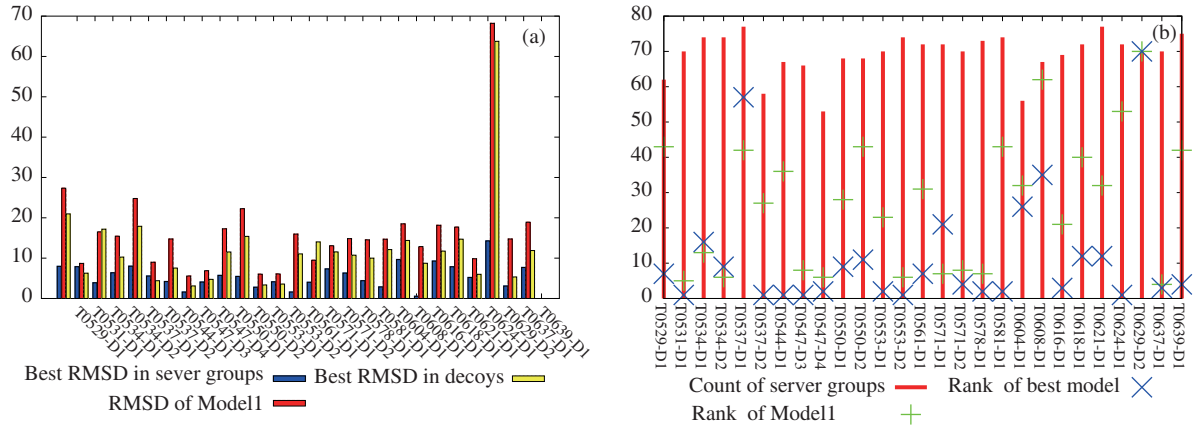

图 5 pacBackbone 与 CASP9 所有 server 组提交结果的比较情况

Figure 5 The RMSD comparisons of 27 FM predictions between pacBackbone and all other servers. (a) RMSD of CASP9 FM target; (b) rank in sever groups

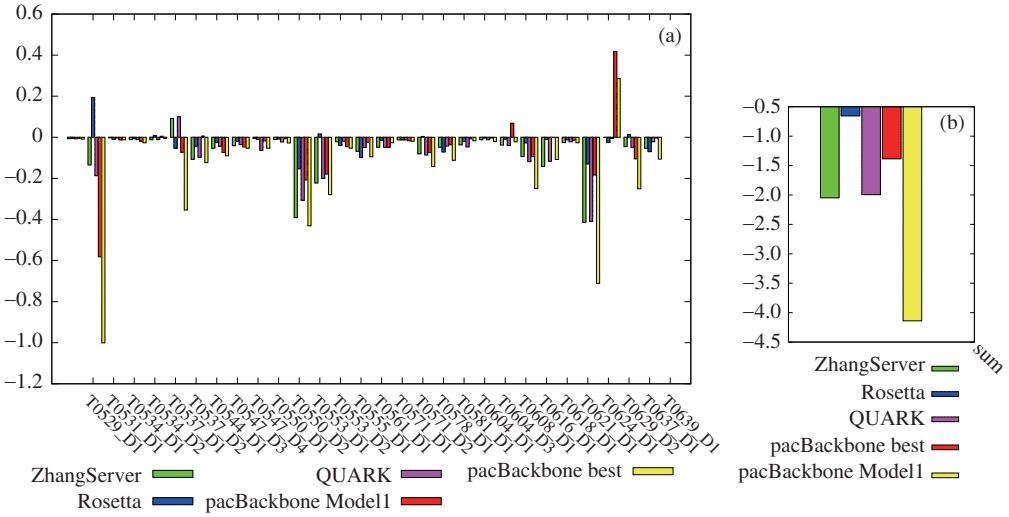

图 6 pacBackbone 与 CASP9 领先系统在所有 FM 目标预测综合 z-score-RMSD 比较

Figure 6 The z-score-RMSD comparison between pacBackbone and the leaders of CASP9. (a) CASP9 FM target z-score (RMSD); (b) sum of z-score

预测问题, 其意义不仅仅在于加速计算, 还在于为该应用问题的解决提供了一种全新的 "*in silico* 实验" 方法.

5.1 pacBackbone 复杂度分析

由于并行算法 pacBackbone 没有对共享的信息素矩阵 $\mathcal{T}$ 进行并发存取控制, 所以 pacBackbone 的时间复杂度是 p 倍于单蚁群算法 1 的复杂度. 算法 1 是启发式搜索算法, 其复杂度是多项式级. 具体地, 对于 n 氨基酸长度的 FM 目标来说: 第 5 行开始的 for 循环的时间复杂度是 $n * p\_cc * 1 * 1 * 200$, 其中 p_cc 是算法常数参数, 通常我们设置为与 n 线性相关. 所以, 该循环的复杂度是 $\mathcal{O}(n^2)$. LocalOptimization($\mathcal{M}_{ib}$) 尝试线性相关 n 次局部替换, 所以时间复杂度是 $\mathcal{O}(n)$. 类似地, 第 19 行的时

间复杂度是 $\mathcal{O}(n)$. 于是, 因为 p_ac, p_ant 也都是与 n 线性相关的算法参数, 所以单蚁群算法 1 的时间复杂度是 $n * p\_ac * \big(n * p\_ant * \mathcal{O}(n^2) + \mathcal{O}(n)\big) + \mathcal{O}(n) = \mathcal{O}(n^4)$.

算法 1 第 5 行开始的循环的空间复杂度主要决定于片段集 $\mathcal{F}$、信息素矩阵 $\mathcal{T}$、蚁群大小等, 除了 $\mathcal{T}$ 是 $\mathcal{O}(n^2)$ 的空间复杂度, 其余都是与 n 线性相关的空间复杂度. 所以总的空间复杂度也是依赖于 $\mathcal{O}(n^3)$.

5.2 三维结构相似度评价标准

评价两个结构之间的三维相似度也是结构预测中的一个独立的困难问题, 很难找到一种评价尺度能既全面又准确地鉴别预测模型与目标模型之间的距离 [29]. 理想的评价标准应是: (1) 全自动的; (2) 能用单一的数值来体现模型的质量; (3) 尽量简单并便于理解; (4) 能适用于不同的结构预测问题, 如: 同源模型比较、自动建模模型比较、折叠模式识别; (5) 能累加计算多个目标的比较; (6) 能被该领域内的大多数研究者所接受. 非常遗憾, 这样的评价标准至今还未发现. 但这并不影响研究者对该问题的关注, 在近二十年间提出许多不同的评价标准. RMSD 一种最经典的评价标准, 它考察两两原子对之间的平均距离, 但 RMSD 对于 10Å 以上结构的区分度不够. GDT_TS 统计了两个结构之间 RMSD 在 1, 2, 4 和 8Å 范围内的原子个数 [32,33], GDT_TS 是迄今为止最为可靠的评价指标 [29], 但是在 CASP 比赛中也会出现专家人工认为最优结构不一致的情况. Q 分数 [29] 比较两个结构的内部距离, 内部距离计算了某残基 Ca 与其它所有残基 Ca 之间的距离. 这样的计算方法克服了基于对齐的相似度计算方法的缺点, 比如: 局部小偏差容易导致分数偏差较大的问题. MAMMOTH(matching molecular models obtained from theory)[34] 在比较局部结构距离时使用了单位矢量 RMSD, 抛弃了常用二级结构相似度, 从而避免了在搜索最大重合结构时对二级结构比较的过于依赖. 其他的评价标准还有 Dali[35], MaxSub[36], TM-score[37] 等等. 正是由于单一数值评价方法的困难, 所以自 CASP3 开始不仅用一组评价分数进行客观评价, 还结合专家的主观评价 [29,38]. 为了尽可能公平的比较 pacBackbone 与其他方法的预测精度, 第一, 本文用多种评价标准进行比较, 将 GDT_TS, Q_{short} 和 Q_{long} 作为 CASP8 预测目标的评测指标, 将 RMSD 作为 CASP9 预测目标的评测标准. 第二, 本文所选的这些标准取自于在近几届 CASP 比赛, 同时也是研究者比较认可的标准, 保证了有足够多的比较对象的数据.

5.3 能量函数并行融合

能量函数是蛋白质三维结构预测算法中的重要目标函数. pacBackbone 中并行蚁群算法的设计, 一方面能够加快运行速度; 另一方面能使每个蚁群拥有独立的能量函数, 并定性互补, 更合理反映天然蛋白质结构的折叠过程. 就本文的案例来说, 这种并行的方法能够探索到 p 倍串行计算资源所采样不到的空间骨架. 由于不同的能量函数作用于蚁群, 使蚁群通过叠代反馈获得了不同的搜索经验. 而 ACO 算法的机制是把搜索经验记录在信息素矩阵中, 所以本文的并行算法让不同蚁群的搜索经验融合到同一个信息素矩阵上, 从而这些搜索经验混杂地共享给所有蚁群. 这种融合方式的定量分析, 因为并行调度的不确定性而体现了强烈的不确定性. 而这恰恰为从头预测问题瓶颈问题之一 (即多个能量函数都不精确但都有用) 提供了一种崭新的解决方案. 相对于传统的实验方式来解构蛋白质骨架, 以及串行或简单并行计算预测, 本文的方法是一种无可替代的 "*in silico* 实验" 方法.

从计算科学来说, 本文需要解决的是多目标优化问题. 不管在搜索方法上如何进步, 最终都会遇到一个难点: 如何评价多个目标函数值冲突的解. 最精确地评价需要引入相应应用问题的主观层面的度量, 在纯数值的层面无法彻底解决. 本文的方法是定性地混合多个目标函数对搜索过程的反馈作用,

从而使得最终的解集合体现了多个目标函数的联合影响. 另一方面, 我们不知道这种能量函数并行组合的方法对构象质量的提高是否总是正面的. 总之, 要想进一步提高构象的质量, 不断优化能量项权重以及能量项本身的准确性总是必要的.

5.4 pacBackbone 的特色与不足

pacBackbone 与现有的从头预测方法有诸多不同. 第一, pacBackbone 能量函数并行融合取代了常用的能量函数串行交替使用方法, 这是一种新的能量函数组合使用的方法. 第二, 构象采样时使用蚁群搜索方法代替常用的 Monte Carlo 搜索, 这种基于蚁群的元启发搜索方式比随机搜索更为有效, 能更快速的找到最优解. 第三, 共享信息素方式使得并行蚁群能更好的传播本种群的优势信息, 提高多个蚁群的搜索质量. 这些特征使得 pacBackbone 能得到质量更高的预测精度.

另一方面, pacBackbone 还有一些有待提高的地方. pacBackbone 可以定性地融合了多个能量函数, 但是目前还没有定量分析方法, 所以目前的结果是一种与问题实例相关的偶然, 还是具有一定的普适性, 这个问题还没有得到解答. 其次, pacBackbone 目前实现的是一种启发式搜索的多道并行, 但是实际上, 目前构架支持多种启发式搜索的多道并行. 第三, pacBackbone 采用与目标函数无关的、只与解的构型有关的聚类处理技术来发现相似解所聚集的簇, 最后以簇的代表作为多目标优化的最终结果. 这样的手法虽然是骨架预测问题的生化模型使然, 但是, 这对于解决其他多目标优化问题的最终解的主观选择问题, 无疑提供了一种有趣的客观 (但是是一种非确定性的) 选择机制. 我们从图 1 和 5 以及在线补充材料 S2, S4 中注意到聚类后模型 1 的质量劣于 decoy 中的最优模型, 这说明我们现有聚类方法从 decoys 中选择模型 1 的能力较弱, 如何从 decoys 中鉴别出最靠近天然结构的预测结构, 是蛋白质骨架预测问题的后续艰难问题.

6 结束语

本文的结果说明, 一方面, 基于共享信息素的并行蚁群优化算法, 融合多个不同的能量函数指导并行搜索, 更好地模拟了天然蛋白质的折叠行为. 经 CASP8/9 中 FM 目标的测试比较, pacBackbone 能够获得较高的骨架质量, 也给预测 FM 目标问题展现了一种新的思路. 另一方面, pacBackbone 在一个算法载体上实现了不同种类搜索策略的融合并行, 对于用非确定性算法解决的目标问题来说也是一种新的独立的方法.

致谢 感谢陈沙沙、缪大俊一起完成本文的实验工作.

参考文献

1 Zhang Y. Process and challenges in protein structure prediction. Curr Opin Struct Biol, 2008, 18: 342–348
2 Baker D, Sali A. Protein structure prediction and structural genomics. Science, 2001, 294: 93–96
3 Rose G D, Flemming P J, Banavar J R, et al. A backbone-based theory of protein folding. PNAS, 2006, 103: 16623–16633
4 Anfinsen C B. Principles that govern the folding of protein chains. Science, 1973, 181: 223–230
5 Protein Stucture Prediction Center. CASP experiments. 2010. Available from: http://predictioncenter.org/

6 Rohl C A, Strauss C E, Misura K M S, et al. Protein structure prediction using rosetta. Method Enzymol, 2004, 383: 66–93

7 Bradley P, Misura K M S, Baker D. Toward high-resolution *de novo* strcture prediction for small protein. Science, 2005, 309: 1868–1871

8 Zhang Y. I-TASSER server for protein 3D structure prediction. BMC Bioinform, 2008, 9: 40

9 Wu S T, Skolnick J, Zhang Y. Ab initio modeling of small prediction by iterative TASSER simulation. BMC Biol, 2007, 5: 17

10 Li W, Wang T, Li E, et al. Parallelization and performance characterization of protein 3D structure prediction of Rosetta. In: 20th International Conference on Parallel and Distributed Processing Symposium. Washington: IEEE Press, 2006. 60–68

11 Tantar A A, Melab N, Talbi E G, et al. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid. Future Gener Comp Sys, 2007, 23: 398–409

12 Raman S, Baker D, Qian B, et al. Advances in rosetta protein structure prediction on massively parallel systems. IBM J Res Dev, 2008, 52: 7–17

13 Calvo J C, Ortega J, Anguita M. Comparison of parallel multi-objective approaches to protein structure prediction. J Supercomput, 2011, 58: 253–260

14 Calvo J C, Ortega J, Anguita M. A hybrid scheme to solve the protein structure prediction problem. Adv Soft Comput, 2010, 74: 233–240

15 Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the travelling salesman problem. IEEE Trans Evolut Comput, 1997, 1: 53–56

16 Dorigo M, Stützle T. Ant Colony Optimization. Cambridge: MIT Press, 2004. 65–114

17 Bowie J U, Eisenberg D. An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function. PNAS, 1994, 91: 4436–4440

18 BakerGroup. Full-chain protein structure prediction server. 2009. http://robetta.org/fragmentsubmit.jsp

19 BakerGroup. Rosetta Software Suite 3.0. 2009. http://www.rosettacommons.org

20 Scheraga H A, Li Z. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. PNAS, 1987, 84: 6611–6615

21 Stützle T, Hoos H H. MAX-MIN ant system. Future Gener Comp Sys, 2000, 16: 889–914

22 Lü Q, Xia X, Qian P. A parallel aco approach based on one pheromone matrix. LNCS, 2006, 4150: 322–329

23 Pan J S, Lü Q, Wang H L. A parallel ant colonies approach for learning bayesian network. Chinese J Comput Sys, 2007, 28: 651–655 [潘吉斯, 吕强, 王红玲. 一种贝叶斯网络结构学习的并行 aco 方法. 小型微型计算机系统, 2007, 28: 651–655]

24 Guo H, Lü Q, Wu J, et al. Solving 2D HP protein folding problem by parallel ant colonies. In: Conference on BioMedical Engineering and Informatics. Tianjin: IEEE Press, 2009. 1525–1530

25 Tsutsui S. Parallel ant colony optimization for the quadratic assignment problems with symmetric multi processing. In: 6th Biannual International Conference on Ant Colony Optimization and Swarm Intelligence. Brussels: Springer, 2008. 363–370

26 Yang Z, Jeffrey S. SPICKER: a clustering approach to identify near-native protein folds. J Comput Chem, 2004, 25: 865–871

27 Huang X, Lv Q, Qian P D. An exemplar selection algorithm for protein structures clustering. Acta Autom Sin, 2011, 37: 682–692 [黄旭, 吕强, 钱培德. 一种用于蛋白质结构聚类的聚类中心选择算法. 自动化学报, 2011, 37: 682–692]

28 Frey B J, Dueck D. Clustering by passing messages between data points. Science, 2007, 315: 972–976

29 Ben-David M, Noivirt-Brik O, Paz A, et al. Assessment of CASP8 structure predictions for template free targets. Proteins, 2009, 77: 50–65

30 DeLano W L. The PyMOL User's Manual. San Carlos: DeLano Scientific LLC, 2002. 6–38

31 Raman S, Vernon R, Thompson J, et al. Structure prediction for CASP8 with all-atom refinement using rosetta. Proteins, 2009, 77: 89–99

32 Zemla A, Venclovas C, Moult J, et al. Processing and analysis of CASP3 protein structure predictions. Proteins, 1999, 3: 22–29

33 Ginalski K, Grishin N V, Godzik A, et al. Practical lessons from protein structure prediction. Nucleic Acids Res, 2005, 33: 1874–1891

34 Ortiz A R, Strauss C E M, Olmea O. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Sci, 2002, 11: 2606–2621
35 Holm L, Sander C. Dali: a network tool for protein structure comparison. Trends Biochem Sci, 1995, 20: 478–480
36 Siew N, Elofsson A, Rychlewski L, et al. MaxSub: an automated measure for the assessment of protein structure prediction quality. Bioinformatics, 2000, 16: 776
37 Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins, 2004, 57: 702–710
38 Jauch J, Yeo H C, Kolatkar P R, et al. Assessment of CASP7 structure predictions for template free targets. Proteins, 2007, 69: 57–67

A parallel ant colonies approach to *de novo* prediction of protein backbone in CASP8/9

WU HongJie [1,3], LV Qiang [1,2*], WU JinZhen [1], HUANG Xu [1], LUO XiaoHu [1] & QIAN PeiDe [1,2]

1 *School of Computer Science and Technology, Soochow University, Suzhou* 215006*, China;*
2 *Jiangsu Provincial Key Lab for Information Processing Technologies, Suzhou* 215006*, China;*
3 *School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou* 215009*, China*
*E-mail: qiang@suda.edu.cn

Abstract Predicting the three-dimensional structures of proteins from amino acid sequences with only a few remote homologs, or *de novo* prediction, remains a major challenge in computational biology. The *de novo* modeling of the protein backbone is the prerequisite stage of a protein structure prediction process. Using a parallel ant colony optimization based on sharing one pheromone matrix, this paper proposes a parallel approach to predicte the structure of a protein backbone. The parallel approach combines various sources of energy functions and generates protein backbones with the lowest energies jointly determined by the various energy functions. All the free modeling targets in CASP8/9 are used to evaluate the performance of the method. For 13 targets in CASP8, two out of the predicted model1s selected by our approach are the best of the published CASP8 results, and seven out of the model1s are ranked in the top 10. For 29 targets in CASP9, 20 out of the best models from our predictions are ranked in the top 10, and 11 out of the model1s are ranked in the top 10. The solution described in this paper mimics the nature behavior of native protein folding by simultaneously minimizing the values of multiple energy functions. It also provides a general framework to combine different search strategies in parallel platform, which is a novel approach to solving the similar optimization problems with non-deterministic algorithms.

Keywords protein backbone, *de novo* prediction, protein folding, parallel algorithms, heuristic algorithms

WU HongJie received his M.S. degrees in computer science from Soochow University, Suzhou in 2005. He is currently a Ph.D. candidate in Soochow University. His research interests include bioinformatics, meta heuristics search, parallel and distributed computing.

QIAN PeiDe received his B.S. degrees in computer science from Nanjing University in 1982. He is currently a Professor at School of Computer Science and Technology, Soochow University. His research interests include Chinese information processing, distributed computing, and operating system.

LV Qiang graduated from Soochow University, Suzhou in 1988. He received the M.S. degree from China Eastern Institute of Technology in 1991 and the Ph.D. degree from Soochow University in 2006. He is currently a Professor at School of Computer Science and Technology, Soochow University. His research interests include bioinformatics, meta heuristics search, parallel and distributed computing.