一种用于蛋白质结构聚类的聚类中心选择算法

黄 旭 [1] 吕 强 [1,2] 钱培德 [1,2]

摘 要 提出一种对蛋白质结构聚类中心进行选择的算法. 聚类是蛋白质结构预测过程中必不可少的一个后处理步骤, 而目前在蛋白质结构预测中常用的属性阈值 (Quality threshold, QT) 聚类算法依赖于由经验得出的聚类半径; 其他聚类算法, 如近邻传播 (Affinity propagation, AP) 聚类算法也存在影响聚类分布的参数. 为克服对主观经验参数的依赖, 本文提出一种聚类中心选择算法 (Exemplar selection algorithm, ESA), 用于对不同参数下的聚类结果进行分析, 从而选择最佳聚类中心, 进而确定聚类半径等经验参数. 该算法在真实蛋白质结构数据集上进行了实验, 在未知经验参数情况下选择出最佳聚类中心, 同时也为不同聚类算法寻找适合相应数据集的客观聚类参数提供了支持.

关键词 蛋白质结构, 聚类, 属性阈值, 近邻传播, 聚类中心选择

DOI 10.3724/SP.J.1004.2011.00682

An Exemplar Selection Algorithm for Protein Structures Clustering

HUANG Xu[1] LV Qiang[1,2] QIAN Pei-De[1,2]

Abstract This paper proposes an exemplar selection algorithm (ESA) for protein structures clustering, which is a necessary post-processing step for protein structure prediction. The widely-used quality threshold (QT) algorithm in protein structure prediction depends on clustering radius derived from experience, which also affects clustering distribution in other widely-used clustering algorithms such as affinity propagation (AP). The proposed exemplar selection algorithm can analyze clustering results, choose the best exemplar, and confirm clustering parameter such as clustering radius. Experimental results on real protein structure predictions confirm the effectiveness of our exemplar selection algorithm, which can choose the best exemplar with no experience parameter, and can find the best parameter fitting for data set.

Key words Protein structure, clustering, quality threshold (QT), affinity propagation (AP), exemplar selection

蛋白质结构预测是指由蛋白质的氨基酸序列出发, 采用计算的方法预测该序列所对应的最合理三维结构. Anfinsen 于 1973 年提出的蛋白质天然结构具有最低自由能的热力学原理[1] 是指导蛋白质结构预测的重要理论基础. 因此, 如何发现具有最低自由能的状态成为蛋白质结构预测中的关键问题. 在目前的预测过程中, Anfinsen 热力学原理以能量函数的形式体现, 预测人员在能量函数指导下构建一条序列所对应的三维结构. 一个系统的能量降低, 意味着系统状态更加稳定, 因此天然状态应该是一种相当稳定的状态. 然而, 文献 [2] 指出, 蛋白质结构预测问题受限于两个方面: 一方面, 由于有机分子及其内部微粒之间关系的复杂性, 目前的能量函数并不能精确反映分子系统的能量状况, 只是在某个侧面的近似; 另一方面, 最低自由能状态是由能与熵共同竞争所导致的一种平衡状态, 而分子能量减少仅是导向最低自由能的因素之一. 因此, 在人们未能完全认识最低自由能精确度量的情况下, 虽然能量函数成为寻找最低自由能构象不可或缺的指导信息, 但仅靠能量函数仍不足以探寻最低自由能构象. 所以在实践过程中通常要用到聚类技术.

由于生理环境下活性蛋白质具有动态构象, 也就是说蛋白质的天然构象并非固定状态, 因而在蛋白质一维氨基酸序列所对应的各种可能结构中, 最低自由能状态具有最大的出现概率. 因此, 最低能量状态与最低自由能状态之间并非严格对应. 由于能量函数缺乏对位形熵因素的考虑, 实际上是无法评估这种状态的, 需要在预测出最终结构之后的后处理步骤中对上述缺陷加以弥补. 正是由于自由能的分布特性以及发现准确能量函数的困难性, 结构预测人员通常生成尽可能多的候选结构 (Decoys). 所谓蛋白质结构聚类, 就是从这组候选结构中通过聚类的方法发现最具代表性的结构类, 近似认为类中结构具有相同的位形熵, 进而确定最具代表性的类内中心作为最低自由能状态所对应的结构. 普遍认为最大聚类的聚类中心更接近于最低自由能状态[3].

本文所指的聚类中心, 更多的是面向应用问题的概念, 并非简单的类几何中心. 就蛋白质结构预测这一具体应用问题而言, 聚类中心有多种选择标准.

收稿日期 2010-09-07 录用日期 2010-12-27
Manuscript received September 7, 2010; accepted December 27, 2010
国家自然科学基金 (60970055) 资助
Supported by National Natural Science Foundation of China (60970055)
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
1. School of Computer Science and Technology, Soochow University, Suzhou 215006 2. Jiangsu Provincial Key Laboratory for Computer Information Processing Technology, Suzhou 215006

传统的聚类几何中心自然是常见的选择标准. 另外一种观点是, 聚类算法确定聚类划分所依据的是蛋白质候选结构之间的相似关系, 而从同一个类中选择更接近天然状态的候选结构, 似乎从能量的角度进行区分更为合理[4−5]. 目前更倾向于面向蛋白质结构的聚类分析应该充分利用已知的生物学知识[6]. 因此, 蛋白质结构聚类后的聚类中心选择问题, 一直是一个难而未决的问题. 国际会议 CASP (The Critical assessment of techniques for protein structure prediction)[7] 一直被认为是蛋白质结构预测领域的 "奥林匹克" 大会. 在 CASP 历届比赛中, 对于蛋白质结构预测都采用首先生成候选结构集合, 然后通过聚类寻找聚类中心的方法来确定最终结构. 然而, 在序列所对应天然结构公布之后的验证表明, 最初所提交的结构往往不是候选结构集合中的最优结构.

目前生物数据聚类中常用的聚类算法是属性域值 (Quality threshold, QT) 算法[8], 该算法的聚类效果依赖于由用户经验所指定的聚类半径. 这是传统密度聚类算法的共同缺点[9]. 在其他聚类算法中, 影响聚类分布的参数往往也与解决领域问题的经验有关. 比如在近邻传播 (Affinity propagation, AP)[10] 聚类算法中, 也需要由用户指定一个合适的 Preference 参数, 以便获得合理的聚类划分. 在缺乏相关经验的情况下, 如何完成高质量的聚类是一项值得探讨的工作. 本文采用的方案是通过在一个参数范围之内进行多次聚类尝试, 然后依据聚类结果所反映出来的聚类中心的性质选择最佳聚类中心, 同时也可获得相应的最佳聚类参数. 本文将以 QT 算法以及 AP 算法为基础, 探讨蛋白质结构聚类中的聚类中心选择问题.

1 聚类算法对类的刻画

如何区分不同的类是聚类算法的关键步骤, 相关参数设置也各不相同. QT 聚类算法是针对生物数据进行聚类的一种算法. 该算法简单高效, 广泛用于生物领域中的基因聚类、蛋白质结构聚类中. 比如, 蛋白质结构预测领域著名的 Rosetta 平台在处理蛋白质 Decoys 集合时, 就采用了该算法[11]. 然而, 该算法依赖于由用户主观经验所指定的聚类半径, 聚类效果与聚类半径密切相关. 而 AP 算法可以更快地处理大规模数据, 得到较好的聚类结果[12]. 该算法虽未明确采用 "聚类半径" 的概念, 但其 Preference 参数仍然对最终的聚类分布有直接影响.

1.1 QT 算法及其聚类半径

与 K-means 算法相比, QT 算法需要更多的计算时间, 然而它不需要事先指定类的数目. 此外, QT 算法是一种确定性的算法. 其基本思路是:

1) 用户选择一个最大聚类半径;

2) 为每一个数据点创建候选类, 候选类中包含了半径范围之内的所有数据点;

3) 把具有最多点数的候选类作为第一个类, 并从集合中删除该类中的点;

4) 在剩余点中递归处理.

其中, 聚类半径是 QT 聚类算法中特有的一个概念. 聚类半径定义如下:

定义 1 (聚类半径). 聚类半径是 QT 聚类算法中用于衡量两个数据点能否划归同一个类的阈值. 对于数据点 D_i 与 D_j, 针对事先指定的聚类半径 r_0, 如果 $|D_i - D_j| < r_0$, 则表明可以将数据点 D_j 划归以 D_i 为聚类中心的类中.

QT 算法采用这样一个生硬的参数确定类的范围, 而不能识别实际数据集合中合理的类别边界, QT 所识别出的聚类分布与实际数据的自然分布往往有较大差异. 然而, 在处理蛋白质、基因等大规模生物数据的实际应用场景下, 算法执行的效率变得格外突出, 而且, 实际经验表明, 针对这些生物数据, 只要确定了合适的聚类半径, 仍会取得较高质量的聚类结果. 这是 QT 算法被广泛用于生物数据聚类的一个原因. 具体的 QT 算法如下[8]:

算法 1. QT clustering algorithm

1) Procedure QT_Clust(G, d)

2) if ($|G| \leq 1$) then output G, else do /*Base case*/

3) for each $i \in G$

4) set $flag$ = True; set $A_i = \{i\}$ /*A_i is the cluster started by i*/

5) while (($flag$ = True) and ($A_i \neq G$))

6) find $j \in (G - A_i)$ such that $diameter(A_i \cup \{j\})$ is minimum

7) if ($diameter(A_i \cup \{j\}) > d$)

8) then set $flag$ = False

9) else set $A_i = A_i \cup \{j\}$ /*Add j to cluster A_i*/

10) identify set $C \in \{A_1, A_2, \cdots, A_{|G|}\}$ with maximum cardinality

11) output C

12) call QT_Clust(G, d)

1.2 AP 算法及其偏向参数

聚类半径是 QT 算法基于生物数据聚类的特点, 为简化聚类过程而规定的一个阈值参数. 该阈值在整个聚类过程中是静态的. 在其他的一些聚类算法中, 如果也用聚类半径这个概念来衡量的话, 聚类半径往往是不固定的; 然而, 这些算法必然存在另外一些固定的参数用以区分不同的类. 比如

在 AP 聚类算法[10] 中有一个重要的参数是 Preference, 文献 [13] 称之为偏向参数. 该参数表明了数据点选择自己作为聚类中心的初始意愿. Preference 值越大, 将导致最终聚类结果中类的数目越多, 相应地, 每一个类中成员的数目会更少. Preference 实际上反映了一种隐性的 "聚类半径".

AP 算法以数据点对之间的相似性作为输入, 在数据点之间交换消息, 经过若干次迭代之后, 消息的传播达到一种稳定状态. 在这样一个过程中, 有两种类型的消息: 一种被称作 responsibility (记作 $r(i,k)$), 另一种是 availability (记作 $a(i,k)$). 算法的核心步骤是这两个消息的交替更新过程. 消息 availability 初始化为 0: $a(i,k)=0$, 而 $r(i,k)$ 和 $a(i,k)$ 由下列公式进行迭代计算:

$$r(i,k)\leftarrow s(i,k)-\max_{k'\text{s.t.}k'\neq k}\{a(i,k')+s(i,k')\} \quad (1)$$

$$a(i,k)\leftarrow \min\{0,r(k,k)+\sum_{i'\text{s.t.}i'\notin\{i,k\}}\max\{0,r(i',k)\}\} \quad (2)$$

$$a(k,k)\leftarrow \sum_{i'\text{s.t.}i'\notin\{i,k\}}\max\{0,r(i',k)\} \quad (3)$$

其中, $s(i,k)$ 是数据点 i 与 j 之间的相似性. 作为算法输入的相似性矩阵 s 在初始状态缺失了对角线元素 $s(k,k)$, 由算法采用 Preference 进行初始化, 默认情况下 Preference 取矩阵 s 中元素的中值 (median). 在消息不断传播的过程中, availabilities 和 responsibilities 共同确定聚类中心. 对于点 i, 具有最大值的 $a(i,k)+r(i,k)$, $i,k=1,2,\cdots,n$, 指出了点 i 属于中心点 k 所对应的类 (此时 $k\neq i$), 反之, 若 $k=i$, 则表明点 i 将选择自己作为其所在类的中心.

这样一个过程中, 消息 $r(i,k)$ 由数据点 i 发出, 并由潜在的聚类中心 k 接收, 该消息反映了点 k 主动作为点 i 所在类中心的意愿. $r(i,k)$ 值越大, 说明作为类中心的意愿越强烈. 而消息 $a(i,k)$ 由潜在的聚类中心 k 发出, 并由数据点 i 接收, 反映了点 i 选择点 k 作为自己所在类中心的可能性. 该值越大, 说明选择对方作为聚类中心的意愿越强烈. 因此, 这两种消息实际上反映了在数据点集合中 "自荐" 与 "选举" 的交互. 消息值也是随着迭代而不断变化的, 最终的平衡状态应该呈现一种合理的聚类划分. 文献 [12] 称 $r+a$ 为决策矩阵, $s+a$ 为潜力矩阵, 更加清晰地注解了 AP 的过程. 通过这种方式, AP 算法避免了聚类半径所带来的硬性划分而使类别之间的界限更加自然. 然而, 具体的划分结果仍然受到算法偏向参数的影响. 而且, 如何确定偏向参数以使算法产生最优聚类结果是一个难题[13].

1.3 类别区分参数对聚类结果的影响

根据前面对两个经典聚类算法 QT 和 AP 的分析, 认为聚类算法中总会存在一个用以区分不同类的参数, 该参数未必与类的大小直接相关, 而 QT 算法中的聚类半径是最简单、最直接的一个参数. 本文正是从 QT 入手, 对这种参数进行剖析, 进而把研究结论扩展到其他聚类算法中.

蛋白质结构数据分布极不均匀. 在这种情况下, 采用静态聚类半径的聚类算法未必能获得划分良好的聚类结果[14]. 如果静态半径硬性划分的聚类分布与数据的自然分布有所偏差, 将导致尽管类的密度很大, 但类中的成员并不适合划归同一个聚类. 这种现象表明聚类半径的选取不是最佳的, 反映了 QT 算法依赖硬性聚类半径的弊端. 在被聚类数据分布未知的情况下, 如何确定半径 r 的确是一个难题. 同样, 在其他一些未采用聚类半径的聚类算法中, 如何确定用以区分不同类的参数也是不容易的. AP 算法的默认 Preference 取数据集相似性的中值. 本文认为相似性中值未必能真正合理反映数据集合的分布特征. 获得更准确的 Preference 与 QT 算法中获取更合理的聚类半径具有类似的意义.

研究在聚类算法中采用自适应的聚类参数也是一种解决思路[9, 13−14]. 然而, 对蛋白质结构聚类而言, 算法效率是首要考虑的因素. 因此, 本文认为在不改变已有聚类方案的情况下, 确定合适的聚类参数对蛋白质结构聚类更有现实意义.

2 聚类中心选择算法

针对 QT 算法依赖于经验半径的不足, 本文认为, 在不易确定聚类半径的情况下, 可在一定半径范围内进行聚类, 获得多组聚类信息, 然后分析这些聚类信息以进行择优选择. 本文提出一种聚类中心选择算法, 用于对聚类信息的分析处理. 算法的结果即最佳聚类中心; 同时通过该聚类中心可获得相关的聚类算法参数信息.

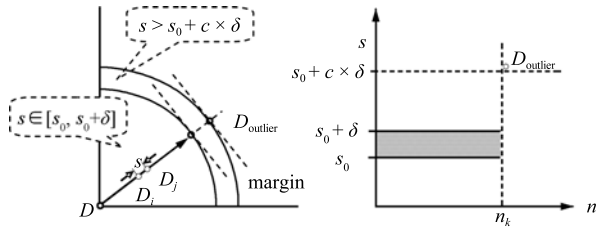
2.1 基本思路

结合对 QT 算法的理解, 本文认为, 在蛋白质结构聚类中, 确定最佳聚类中心的关键是寻找聚集程度最高的一个子集. 聚集程度定义如下:

定义 2 (聚集程度). 聚类数据集合 I 中, 在以某个数据点 D 为中心、包含 I 内所有数据的最小球 Ω_I (球面记作 Φ_I, 半径记为 R) 内, 从 D 开始, 沿着球面任意一个法线方向 $\boldsymbol{r}_k$, 相邻数据点 D_i 和 D_j 之间总是以相对恒定的步长 s 均匀向外扩展 ($s\in[s_0, s_0+\delta]$), 直到找不到下一个相邻的数据点或步长 s

发生较大变化时 ($s > s_0 + c\cdot\delta$) 为止, 在这样一个步进 (walking) 过程中, 走过的步数 n_k 为方向 $\boldsymbol{r}_k$ 上的跨越距离. 其中, δ 是 s 的变化容忍度, s_0 是最小步长, c 是常数. 当步长 s 发生超出容忍度范围的较大变化时, 称找到了一个离群点 (outlier)[15]. 此时点 D_i (记作 D^k) 称作在 $\boldsymbol{r}_k$ 方向上的边界点. 所有的点 D^k 构成一个曲面 Φ^*, 曲面 Φ^* 内的数据点组成以点 D 为聚集中心的一个子集. 而所谓 D 点的聚集程度, 是指在所有法线方向上跨过的平均距离, 即聚集程度 $G = \frac{1}{m}\sum_{k=1}^{m} n_k$ ($k = 1, 2, \cdots, m$, m 为所有法线方向离散化的取值数目).

上述定义中经离散化之后, "法线方向" 也就不再是单纯几何意义上的 "直线", 而是一个区域. 所谓 D, D_i, D_j, D_{outlier} "在法方向上" 实际上是指在离散化之后的一个 "法方向区域" 内. 聚集检测过程如图 1 所示.



(a) 二维空间中的聚集检测　(b) 检测过程中步长的变化

(a) Mass detecting in two dimensional space　(b) Step variation in mass detecting

图 1　聚集检测过程示意图

Fig. 1　Schematic diagram of mass detecting

步长 s 超出 $[s_0, s_0+\delta]$ 意味着脱离聚集中心, 或者说检测到离群点是步进过程结束的标志. 该状态在图 1 (a) 中呈现出一个较宽的 margin, 在图 1 (b) 中呈现出 s 在值域上变化的峰值. 显然, 数据聚集程度隐含了两个方面的分布特征: 相邻数据点之间的距离以及紧密相邻数据点的分布范围. 数据聚集程度越高, 表明相邻数据点之间的距离 s 越小, 紧密相邻的数据点分布范围也越广 (数据点 D 到聚类边界的距离越长).

法线方向离散化的取值数目 m 取决于法方向离散化的粒度 p. 可采用如下算法确定 m: 设法方向与球面 Φ_I 交点的集合为 Ψ (初始状态 $\Psi = \varnothing$): 1) 计算球面 Φ_I 上任意点 $D^i(X_i^1, X_i^2, \cdots, X_i^d)$ 处的法方向 $\boldsymbol{r}_i = (\frac{\partial\Phi_I}{\partial X^1}, \frac{\partial\Phi_I}{\partial X^2}, \cdots, \frac{\partial\Phi_I}{\partial X^d})$, $\Psi \leftarrow \Psi \cup D^i$; 2) 在 D^i 所在切面 α_i 上任取一个方向 $\boldsymbol{t}_0$ ($\boldsymbol{t}_0 \perp \boldsymbol{r}_i$); 3) 以上一步确定的方向为基准, 在切面 α_i 上确定由 D^i 出发的其他方向 $\boldsymbol{t}_j$ ($j = 1, 2, \cdots, p-1$), 满足 $\boldsymbol{t}_j$ 与 $\boldsymbol{t}_{j+1}$ 的夹角为 $2\pi/p$; 4) 在所确定的每一个方向 $\boldsymbol{t}_j$ 上以步长 $R\times 2\pi\cdot R/p$ 沿球面 Φ_I 找到另一个点 $D^{i'}$, 如果对于所有 D^Ψ ($D^\Psi \in \Psi$), 都有 $|D^{i'} - D^\Psi| > R\times\sqrt{2(1-\cos(2\pi/p))}$, 则计算该点处的法方向 $\boldsymbol{r}_{i'}$, $\Psi \leftarrow \Psi \cup D^{i'}$, 否则终止算法; 5) 用 $\boldsymbol{t}_0$ 表示 $\boldsymbol{t}_j$, 重复第 3) ~ 4) 步, 至此, 找到的所有法方向数目为 $m = \text{size}(\Psi)$.

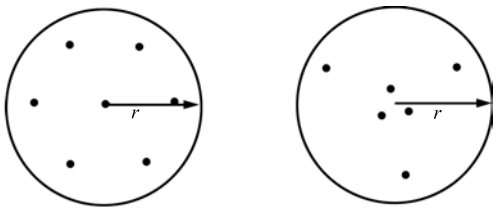
实际操作中难点在于离散化粒度 p 的选择. 在粒度 p 过小的情况下, 在特定的法方向上可能出现找到离群点而其实并未脱离聚集中心的情况. 当然, 这种情况并不影响定义的理论表达.

对蛋白质结构预测这一具体应用而言, 虽然蛋白质结构信息的描述是一个高维 ($10^3 \sim 10^4$) 数据, 但在蛋白质结构聚类中, 输入信息实际上是候选结构集合的距离分布, 经过距离度量计算之后, 维数得以大大降低. 聚集程度不仅可很好地适用于蛋白质结构聚类问题, 也可在蛋白质结构预测或蛋白质对接过程中用于对结构合理程度的判断. QT 算法之所以不能良好识别真实数据的自然分布, 就是因为 QT 仅仅把聚类半径作为一个阈值, 而忽略了从聚类中心到聚类边界的步进检测过程. 上述聚集程度的定义中正好描述了这样一个过程, 弥补了 QT 算法的缺陷, 将会最大程度地拟合聚类算法的聚类划分与实际数据的自然聚类划分.

聚集程度与聚类密度是不同的. 聚类密度在许多聚类分析中都有所应用. 聚类密度定义如下:

定义 3 (聚类密度). 聚类密度是在一定的聚类参数下, 类中成员的个数与聚类半径 d 次方 (d 为维数) 的比值. 设聚类半径为 r, 类 I_i 中的成员个数为 n_i, 则该类的密度 $\rho_i = c\cdot n_i / r^d$, c 为常数.

聚类密度反映的是数据聚集区域的平均分布情况, 因此, 聚集程度与聚类密度虽然相关, 但二者之间并非存在必然的对应关系. 某个类的密度大, 并不意味着聚类中心的聚集程度高. 图 2 所示为聚类密度与聚集程度示意图.



(a) 数据点多, 且较分散　(b) 数据点少, 且较聚集

(a) Dada points are more and scattered　(b) Data points are less and centralized

图 2　聚类密度与聚集程度示意图

Fig. 2　Schematic diagram of clustering density and mass degree

图 2 (a) 和 2 (b) 两个聚类的面积相同, 由于图 2 (a) 中数据点更多 (图 2 (a) 中 7 个点, 图 2 (b) 中 6 个点), 显然图 2 (a) 的密度大; 而图 2 (b) 中心附

近的聚集程度更高. 也就是说, 图 2 (b) 中心附近的数据点比图 2 (a) 中心的数据点具有更高的出现概率, 符合通常对自由能的理解, 因此, 对蛋白质结构聚类而言, 倾向于将图 2 (b) 的中心点作为最优聚类中心. 当然, 上述分析是以聚类参数 r 为基础的, 如果采用另外一个更为合适的 r, 聚类的结果也很有可能是图 2 (b) 点附近的密度更大. 但关键是在不了解数据集合分布的情况下, 如何确定 r 是一个难题. QT 算法采用了一个基于经验的 r 值, 对聚类结果的衡量只考虑到了类的大小 (在半径相同的情况下, 类的大小与类的密度成正比, 因此实际上考虑的是密度), 这是 QT 算法受限于经验聚类半径的原因之一.

本文将数据点附近的聚集程度与聚类算法结合起来进行考虑, 主要是基于以下两个原因: 1) 目前常用的聚类算法并未考查各数据点附近的聚集程度, 例如 QT 算法仅考虑特定半径下的平均密度, 而不能反映 "聚集" 这一动态过程; 2) 即使基于聚集程度概念来设计聚类算法, 也仅是对 QT 算法的部分改进. 而将聚集程度与聚类算法结合考虑, 可在已有聚类算法的基础上, 针对特定的数据集, 研究符合聚集程度的聚类参数, 有利于充分发挥已有聚类算法的优势, 减少单纯为计算聚集程度而带来的计算消耗. 基于这种思路, 可以认为, 要准确衡量数据点附近的聚集程度, 需在不同聚类半径下对多次聚类结果进行分析, 以实现定义 2 中所描述的步进过程. 该过程还涉及以下概念:

定义 4 (聚类中心频数). 采用聚类参数 γ_i ($i = 1, 2, \cdots, g$, g 为聚类尝试次数) 对数据集合 I 进行聚类, 得到一个特定的聚类划分 $I^i = \{I^i_1, I^i_2, \cdots, I^i_{v_i}\}$ (v_i 为该次聚类中所产生的类的个数), 每个类的聚类中心分别记为 $C^i_1, C^i_2, \cdots, C^i_{v_i}$. 如果 $\exists j \in \{1, 2, \cdots, v_i\}$, 使得 $D = C^i_j$, 则称数据点 D 在聚类划分 I^i 中处于聚类中心地位. 在所有 g 次聚类尝试中, 设 D 位于聚类中心地位共 p ($p \le g$) 次, 则记数据点 D 处于聚类中心地位频率为 $f = p/g$. 在多次实验聚类尝试次数 g 相同的情况下, 聚类中心频数可直接用 p 表示.

聚类中心频数反映了采用不同聚类参数进行聚类时, 某个数据点保持聚类中心地位的稳定性. 在不同的聚类参数下, 一个数据点聚类中心频数越高, 也就是作为聚类中心的稳定性越高, 表明该点越是处于真实聚类的中心位置. 一个具有较大聚集程度的真实最佳聚类中心应当表现出较好的稳定性. 而对于一些稳定性较差的点, 则在采用不同参数的多次聚类过程中被淘汰.

聚类尝试次数 g 与具体数据集相关, 通过在数据子集上反复进行试探性实验来确定. 如果随着 g 值的变化, 相邻两次聚类尝试的聚类结果变化不大 (通过观察最大 5 个类的聚类中心、类的大小进行判断), 就可以减小 g 值; 反之, 可以增大 g 值. 由于数据子集应该与整个数据集具有相同的分布, 因此可将 g 用于后续的实验. 根据在本文所列数据集上的实验, 本文在这些数据上采用的尝试次数为 5～8 次.

另一个概念是聚类中心位次, 即简单根据类的大小进行排序. 定义如下:

定义 5 (聚类中心位次). 在一定的聚类参数 γ_i 下, 对各聚类中心 $C^i_1, C^i_2, \cdots, C^i_{v_i}$ 按照类的大小从大到小排序, 如果有 $\mathrm{size}(I^i_1) \ge \mathrm{size}(I^i_2) \ge \cdots \ge \mathrm{size}(I^i_{v_i})$, 则聚类中心 C^i_j 的聚类中心位次为 j.

这个概念反映了在一种特定的聚类划分中, 各个类的大小关系. 因为通常认为最大聚类的聚类中心最接近于最低自由能状态. 理论上, 在聚类划分已知的情况下, 仅仅根据聚类中心位次就可以对最低自由能状态作出判断. 而在聚类半径不易确定的情况下, 类的大小可能会对判断造成消极的影响. 如果在参数不同的多次聚类尝试中, 以数据点 D 为中心的类总是处于比较靠前的位次, 说明 D 周围的数据分布是紧密聚集的, 也就是说 D 点周围的聚集程度较高. 这是本文在选择聚类中心时着重考虑的一个因素. 本文算法的最终目的是寻找最佳聚类中心, 最佳聚类中心定义如下:

定义 6 (最佳聚类中心). 在所有 g 次聚类尝试中, 对于 $\sum_{i=1}^{g} v_i$ 个聚类中心, 如果聚类中心 C^* 的聚集程度 $G^* = \max\{G^i_j | i = 1, 2, \cdots, g,\ j = 1, 2, \cdots, v_i\}$, 则 C^* 称为最佳聚类中心.

根据上述定义与分析, 本文认为最佳聚类中心与聚集程度直接相关; 聚类中心频数也从一个侧面反映了聚集程度; 而聚类密度、聚类中心位次对寻找最佳聚类中心也具有积极的作用.

2.2 算法流程

寻找最佳聚类中心, 着重依赖于聚类中心频数与多次聚类尝试中的聚类中心位次之和. 本文认为聚类中心频数越大、聚类中心位次越小, 表明数据点成为最佳聚类中心的可能性越大. 这些属性只有在对多次聚类结果分析的基础上才可获得. 因此, 在每一次聚类实验中, 首先尝试选择不同的聚类参数, 从而获取多个聚类结果. 然后从这多个聚类结果中进行择优. 本文提出的聚类中心选择算法 ESA 就是用于对这些信息进行统计处理. 算法流程如算法 2 所示.

该算法通过考查不同参数下聚类算法获取的聚类中心点的频数、位次, 以及类的大小等信息, 了解聚类中心的稳定性、类的密度等性质, 并以此作为最

佳聚类中心的选择依据. 其中, 步骤 1 考虑到数据点频数, 频数越大反映出稳定性越高; 步骤 2 和步骤 3 通过衡量每个候选中心在不同实验中的位次之和, 综合考虑了候选中心的稳定性及类的密度; 步骤 4 通过计算高频候选中心在同一次实验中的位次之和, 实现了对实验参数质量的考查; 步骤 5 着重考虑类的大小; 而在步骤 6 和步骤 7 中, 如果未达到 $v=1$, 也就是在仅考虑最高频数据点仍不足以区分最佳聚类中心的情况下, 继续考虑次高频数据点, 因为本文认为, 如果在某次聚类尝试中最高频与次高频的聚类中心点都比较稳定, 说明这次聚类尝试要优于其他.

算法 2. Exemplar selection algorithm (ESA)

输入. 各次聚类结果, 包括各次实验中最大 R_{MAX} 个类的大小和聚类中心.

步骤 1. 统计各数据点频数, 设具有最大频数 p 的点共 n 个, 记为: D_i, $i=1,2,\cdots,n$;

步骤 2. 设点 D_i 在 p 次实验中的位次分别为 R_{ij}, $j=1,2,\cdots,p$, 分别计算这些点在不同实验中的位次之和: $T_i=\sum_{j=1}^{p}R_{ij}$, 然后选择 T_i 最小的点作为候选中心点, 设共有 m 个候选中心点, 记作 D'_q, $q=1,2,\cdots,m$;

步骤 3. 选择这些候选中心点中位次 R_{ij} 最小的实验, 设共找到 r 次实验, 记作: E_u, $u=1,2,\cdots,r$;

步骤 4. 计算频数为 p 的所有点 D_i 分别在 r 次实验中的位次之和: $T'_u=\sum_{i=1}^{n}R'_{iu}$, R'_{iu} 是点 D_i 在实验 E_u 中的位次, 如果实验 E_u 中未包含点 D_i, 则令 $R'_{iu}=R_{\mathrm{MAX}}+1$, 然后选择 T'_u 最小的实验, 设有 l 次实验符合要求, 记为: E'_k, $k=1,2,\cdots,l$;

步骤 5. 分别统计这 m 个点 D'_q 在实验 E'_k 中所在类的大小, 记为 S_{qk}, 然后选择 $\max\{S_{qk}|q=1,2,\cdots,m,\ k=1,2,\cdots,l\}$, 设 $\max\{S_{qk}\}$ 所在实验共有 v 次, 记作: E''_t, $t=1,2,\cdots,v$;

步骤 6. 若 $v=1$, 则确定最佳聚类中心为实验 E''_t 中的 D'_q, 算法结束; 否则, 转步骤 7;

步骤 7. 令 $r=v$, 设各数据点次高频数为 p', 令 $p=p'$, 在这 r 次实验中, 频数为 p 的点记为 D_i, 转步骤 4.

输出. 最佳聚类中心.

ESA 算法输入是聚类实验的结果, 算法时间主要消耗在对聚类中心相关属性 (频数、类大小等) 的比较分析上. 因此, 如果在某个数据集上进行 N 次聚类, 每次聚类产生 R_{MAX} 个聚类中心, 则 ESA 算法的复杂度为 $\mathrm{O}(N\times R_{\mathrm{MAX}})$. 而往往在实验参数设计中, N 与 R_{MAX} 值都不大, 因此, 就 ESA 算法本身而言算法复杂度不高. 然而, 由于 ESA 算法执行之前进行了 N 次独立的聚类, 因此确定聚类中心的总的复杂度实际上与所采用的聚类算法相当. 实际上, ESA 算法作为聚类过程的后处理步骤, 其主要贡献在于通过对这些聚类中心的分析, 确定一个最佳选择, 并根据这种选择判断最适合特定数据集合的客观聚类参数.

2.3 应用 ESA 选择聚类中心

ESA 算法的输入是在一定参数范围内的多次聚类结果, 因此需要在算法运行之前确定参数范围, 并对数据集进行多遍聚类. 在理想情况下, 首先尝试选取边界参数 γ_0 与 γ_{N+1}, 使得在参数 γ_0 下每个点自成一类, 而在参数 γ_{N+1} 下所有点聚成 1 类. 然后将区间 $[\gamma_0,\gamma_{N+1}]$ 分为 $N+1$ 等份以获取 $\gamma_1,\gamma_2,\cdots,\gamma_N$ 共 N 个参数, 最后分别采用这 N 个参数进行聚类. 在每一次聚类尝试中, 根据类的大小选择最大的 R_{MAX} 个聚类中心. 其中, N 取值越大, 表明参数步进的粒度越小, 最终的结果分析也越精确, 但聚类的代价也越高. 本文的经验数据表明, 选择 $N=5\sim8$, $R_{\mathrm{MAX}}=10$, 是一个良好的选择.

接下来采用本文提出的 ESA 算法对这 $N\times R_{\mathrm{MAX}}$ 个聚类中心点进行分析, 从中挑选出最低自由能所对应的聚类中心, 然后用该中心与天然结构的相似性来评价性能.

3 算法验证

3.1 实验数据

本文选择了 7 个不同的数据集. 包括 I-TASSER[16] 产生的: 1abv_, 1af7_, 1cqkA, 1csp_, 1dcjA, 256bA 等 6 个数据集, 这些数据下载自 http://zhang.bioinformatics.ku.edu/I-TASSER/decoys/. 同时还选择了数据集 2hq7_, 这是采用 Rosetta 结构预测程序[17] 在本地产生的, Rosetta 版本为 3.0, 运行平台为 1.6 GHz Power(gr) CPU 的 IBM pServer. I-TASSER 是最近几届 CASP 中自由建模类别的姣姣者; Rosetta 在此前的历届 CASP 比赛中都有突出的表现. 这 7 个数据集相关信息如表 1 所示.

表 1　实验用到的数据集

Table 1　Data sets used in experiment

No.	Data Set	Protein	Length	Scale	Generator
1	1abv_-10	1abv_	103	1 250	I-TASSER
2	1af7_-10	1af7_	72	1 250	I-TASSER
3	1cqkA-10	1cqkA	101	2 000	I-TASSER
4	1csp_-10	1csp_	67	1 250	I-TASSER
5	1dcjA-10	1dcjA	73	2 000	I-TASSER
6	256bA-10	256bA	106	2 000	I-TASSER
7	2hq7_-10	2hq7_	141	920	ROSETTA

表中第 2 栏所示为原始数据集的十分之一子集. 由于 I-TASSER 提供的数据集全集尺度过大 (规模

为 10^4), 受限于计算相似性矩阵的时空消耗 (规模为 10^8), 本文未选择数据集全集. I-TASSER 生成十分之一子集采用的方法是: 从 1 号结构开始, 以步长 10 依次进行选择. 实际上, 选择何种子集并不影响最终的结果比较. 况且, 数据集合中的候选结构是随机排列的, 因此可认为子集与全集具有近似的分布特征. 首先计算所选数据子集对应的相似性矩阵 (规模仅为 10^6). 为简化计算, 只生成了上三角矩阵. 由于 AP 算法对相似性矩阵的对称性不敏感, 也就是说, AP 算法要求一个完整的输入矩阵, 因此在用于 AP 算法时对上三角矩阵进行了扩展.

为提高实验效果, 本文对每一数据集进行了三种不同标准的相似性计算, 得出三个不同的相似性矩阵, 然后分别应用于 QT 与 AP 聚类算法. 数据集的分布特征是与测度指标 (metric) 密切相关的. 相同的候选结构集合在不同的测度指标下必然呈现出不同的分布视图, 可视为不同的实验数据. 本文针对每一个数据集采用三种不同的测度指标进行计算, 得到三个不同的相似性矩阵. 因此, 基于上述 7 个数据集实际上可获得 21 组实验数据. 如表 2 所示.

表 2 21 组实验数据相关情况

Table 2 Information of 21 data for experiments

No.	Protein	Metric	No.	Protein	Metric
1	1abv_	RMSD	12	1csp_	GDT_TS
2	1abv_	TM-score	13	1dcjA	RMSD
3	1abv_	GDT_TS	14	1dcjA	TM-score
4	1af7_	RMSD	15	1dcjA	GDT_TS
5	1af7_	TM-score	16	256bA	RMSD
6	1af7_	GDT_TS	17	256bA	TM-score
7	1cqkA	RMSD	18	256bA	GDT_TS
8	1cqkA	TM-score	19	2hq7_	RMSD
9	1cqkA	GDT_TS	20	2hq7_	TM-score
10	1csp_	RMSD	21	2hq7_	GDT_TS
11	1csp_	TM-score			

因此, 准确地说, 本文提到的实验数据是指某种蛋白质候选结构集合在一种特定测度指标下所呈现出来的分布状况 (以相似性矩阵的形式反映). 后续的实验均在上述 21 组实验数据上进行.

3.2 采用的测度指标

为评估两个结构空间上的相似性, 本文采用如下三种测度指标: 均方根偏差 (Root mean square deviation, RMSD)[18], 模板建模评分 (Template modeling score, TM-score)[4, 16, 19] 以及全局距离测试总体评分 (Global distance test_total score, GDT_TS)[20]. 其中, RMSD 是反映两个结构之间相应原子位置差异的量, 在不同的精度下可以采用不同的计算方法, 比如基于骨架原子 (N, C, C_α, O) 的 RMSD 以及基于 C_α 原子的 RMSD 等. 本文采用基于 C_α 原子的 RMSD. 公式如下:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \tag{4}$$

其中, d_i 是不同结构中相对应的第 i 对 C_α 原子之间的距离, 单位为Å, n 是蛋白质氨基酸链上 C_α 原子的数目. RMSD 越小说明两个结构相似性越高. 因此, RMSD 表明的是两个结构之间的差异性, 对于有的聚类算法 (如 AP 算法) 而言, 需转换为相似性作为输入. 本文采用取负数的方法进行转换. AP 算法并未要求将相似性平移至正数空间.

尽管 RMSD 可以指出结构模型的差异, 但在有些情况下, 即使大多数 C_α 原子都重合得很好, 而局部的错误 (如尾部定向错误) 也会导致大的 RMSD 值[4]. 文献 [4, 16, 19] 提出的 TM-score 可以避免这种情况. 公式如下:

$$TM\text{-}score = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{1+\dfrac{d_i^2}{d_0^2}} \tag{5}$$

其中, L 是蛋白质长度, d_i 是第 i 对残基之间的距离, $d_0 = 1.24\sqrt[3]{L-15} - 1.8$. TM-score 值域区间为 $[0, 1]$, 当两个结构完全重合时, TM-score 值为 1.

GDT_TS[20] 定义为在阈值 $D0 = 1, 2, 4, 8$Å 时 GDT 值的平均值, 而 GDT 反映了骨架上相对应的 C_α 原子距离在某个阈值 Di 之内的残基比例. GDT_TS 计算公式如下:

$$GDT\_TS = \frac{1}{4}(N1 + N2 + N4 + N8) \tag{6}$$

其中, Ni 为 $D0 = i$Å 时的 GDT 值[18]. GDT_TS 值域区间也是 $[0, 1]$, 与 TM-score 类似, GDT_TS 值越大说明两个结构的相似性越高. 数据点之间的相似性值通过程序 TM-score[19] 计算, 计算平台为采用 1.6 GHz Power (gr) CPU 的 IBM pServer 平台. 分别计算出三种测度指标下的相似性矩阵, 提供给聚类算法使用. 同时, 对聚类产生的最佳聚类中心, 也通过该程序计算三种测度指标下与天然结构的相似性, 作为评价聚类结果的依据. 因此, 这三种测度指标用于计算候选结构之间的相似性时, 称作相似性度量标准; 而用于评价结果与天然结构之间的相似程度时, 称作评价指标.

3.3 实验结果及分析

本文在采用 1.6 GHz Power(gr) CPU 的 IBM

pServer 平台上实现了 QT 算法. QT 算法的聚类思路简洁, 采用直观的 "聚类半径" 对不同的聚类进行区分. 而在 AP 聚类算法中, 不同的 Preference 值最终导致不同的聚类划分, 从这个意义上讲, Preference 与 QT 算法中的 "聚类半径" 的作用是类似的. 本文提出的算法不仅适用于 QT 算法的结果, 同样也适用于 AP 算法的结果; 而且我们期望能同样适用于其他聚类算法的结果. 本文在采用 3 GHz Pentiun 4 CPU 的 PC 机上进行 AP 聚类.

3.3.1 采用 ESA 算法与未采用 ESA 算法效果比较

对于特定的聚类程序, 通过改变聚类参数, 聚类结果将呈现出一定的差别. 执行 N 次聚类之后, 在每组实验数据上都获得了 N 个最大类的聚类中心, 记为 $D^i_{\max}$ $(i = 1, 2, \cdots, N)$; 由于在每次实验中数值 N 都不相同, 为便于比较, 本文根据结果质量在每一组实验中选择了较好的 5 次实验, 它们最大类的聚类中心分别记为 D_1, D_2, D_3, D_4 和 D_5. 同时在总的聚类结果 ($N \times R_{\mathrm{MAX}}$ 个) 中运用 ESA 算法选择出最佳聚类中心, 记为 D_{ESA}. 然后分别将每个实验数据集上每组实验的 D_{ESA} 与 D_1, D_2, D_3, D_4 和 D_5 进行比较. 对于每组比较数据, 也分别采用上述 3 种评价指标进行评价. 对于表 2 所列出的实验数据, 本文分别采用 QT 算法与 AP 算法进行聚类尝试, 而每一种聚类算法的聚类结果又分别采用 RMSD, TM-score 以及 GDT_TS 进行评价. 因此, 本文实际上进行了 6 组不同的实验, 如表 3 所示.

表 3　6 组不同实验的相关情况

Table 3　Information of 6 experiments

Experiment No.	Clustering algorithm	Metrics
1	QT	RMSD
2	QT	TM-score
3	QT	GDT_TS
4	AP	RMSD
5	AP	TM-score
6	AP	GDT_TS

在每组实验中, 分别对表 2 所示的 21 组实验数据进行聚类, 然后对聚类结果按照上文所述的方法分别选出 6 个聚类中心, 与天然结构进行比较.

由于每组实验的目标蛋白质、聚类算法、评价指标各不相同, 为便于不同实验结果之间的比较, 将实验结果进行了转换, 即把所有的 D_j 值用该值与 D_{ESA} 差距的比例来表示. 公式如下:

$$M_j = \pm\frac{D_j - D_{\mathrm{ESA}}}{D_{\mathrm{ESA}}} \times 100,\ \ j = 1, 2, 3, 4, 5 \quad (7)$$

针对不同的评价指标, 该式取正号或负号: 由于 RMSD 表明的是一种相异程度, 数值越小表明相似性越高, 因此当采用 RMSD 进行评价时, 式 (7) 取 "−" 号; 而 TM-score 或 GDT_TS 反映的是一种相似程度, 数值越大表明相似性越高, 因此当采用这两种指标进行评价时, 式 (7) 取 "+" 号. 这样, 在所有实验中, 如果 $M_j < 0$, 说明 D_{ESA} 优于 D_j; $M_j > 0$, 说明 D_{ESA} 劣于 D_j; $M_j = 0$, 说明二者相同. 上述 6 组实验的结果如图 3 所示.

图 3 (a) ∼ 3 (f) 分别对应表 3 所示的 6 组实验. 即: 图 3 (a) ∼ 3 (c) 是采用 QT 算法的结果; 图 3 (d) ∼ 3 (f) 是采用 AP 算法的结果; 图 3 (a) 和 3 (d) 的评价指标采用的是 RMSD; 图 3 (b) 和 3 (e) 的评价指标采用的是 TM-score; 图 3 (c) 和 3 (f) 的评价指标采用的是 GDT_TS. 图中横坐标为表 2 所示的 21 组实验数据; 纵坐标为通过式 (7) 计算的 M_j 值. 由图中可见, 多数情况下, M_j 位于横轴下方, 即多数情况下, D_{ESA} 要优于 D_j.

接下来, 对每一组实验中 D_{ESA} 与所有 D_j 分别取最优值 (位于最高点) 的比例进行统计. 结果如表 4 所示.

表 4　各聚类中心取得最优值的比例 (%)

Table 4　Ratios of the exemplars which are the best of all (%)

Experiment No.	D_{ESA}	D_1	D_2	D_3	D_4	D_5
1	52.4	7.6	6.7	7.6	2.9	5.7
2	57.1	4.8	5.7	10.5	4.8	7.6
3	52.4	7.6	5.7	12.4	5.7	5.7
4	57.1	6.7	9.0	16.9	7.9	4.5
5	33.3	5.6	10.1	13.5	9.0	5.6
6	47.6	6.7	12.4	15.7	7.9	5.6

表 4 各行数据分别来自图 3 (a) ∼ 3 (f). 如第 1 行数据所示, 在第 1 组 21 次实验中, D_{ESA} 有 11 次取得最优值, 占 52.4 %. 而 D_1, D_2, D_3, D_4 和 D_5 分别有 8, 7, 8, 3, 6 次取得最优值, 同时, 由于这 5 个最大聚类中心来自不同的聚类尝试, 对应于不同的聚类参数, 而通常不能确定哪个聚类参数更有可能获得最佳值, 只能认为这些聚类尝试获得最佳值的概率是相同的 (在本文根据聚类质量优选 5 个聚类中心进行比较的情况下, 概率分别为 20 %; 实际上在未知聚类质量的情况下, 由于做了 N 次聚类, 概率应该仅为 $1/N$), 等价于在这 5 次聚类的全部候选中心中进行选择. 因此, 这些数据在 105 个候选中心中分别占 7.6 %, 6.7 %, 7.6 %, 2.9 % 和 5.7 %. 换句话说, 在每一组实验中, ESA 算法在给出一个最

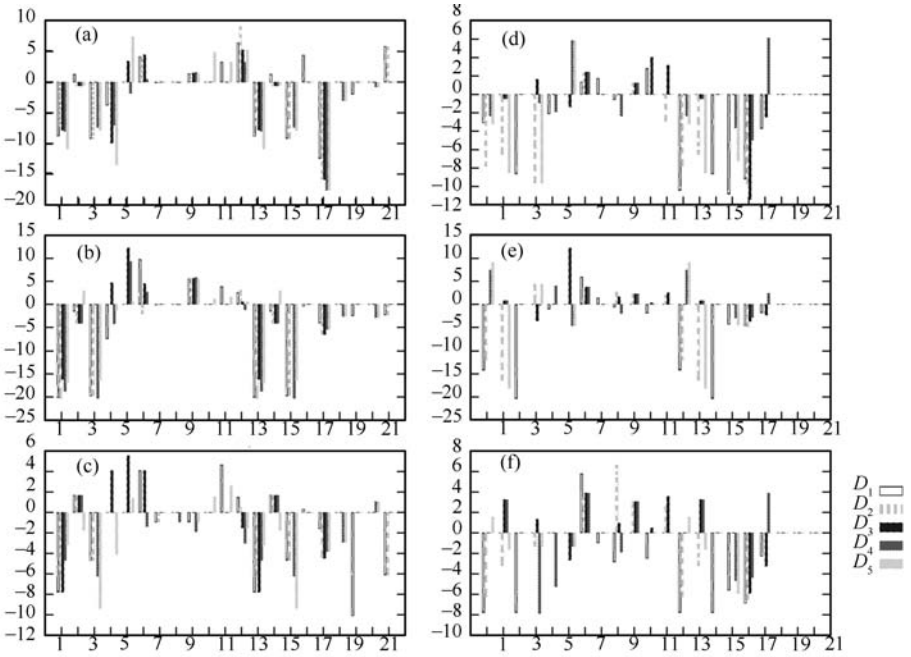


图 3 D_{ESA} 与其他聚类中心的比较

Fig. 3 Comparison of D_{ESA} and other exemplars

佳聚类中心的同时, 还减少了从多次聚类尝试中进行选择的不确定性. 需要说明的是, 由于 AP 算法的收敛性尚未得到很好的证明, 对于事先选择的一些聚类参数存在不收敛的问题[9]. 本文在应用 AP 算法进行聚类的第 4～6 组实验中分别排除了 16 条不收敛数据, 总的候选中心减少为 89 个. 因此在表 4 中前 3 行 (采用 QT 算法) 与后 3 行 (采用 AP 算法) 所对应的 3～7 列在计算百分比时基数有所不同 (分别为 105 与 89).

表 4 反映出, 本文的最佳聚类中心选择算法无论是针对采用 "聚类半径" 作为类别区分参数的 QT 算法还是针对采用隐性 "聚类半径" 的 AP 算法, 都能有效减少选择的不确定性, 使得选中最佳聚类中心的可能性大大提高. 值得注意的是, 表 4 中前 3 行与后 3 行分别针对相同聚类结果信息进行分析, 采用不同评价指标的分析结果略有差别, 实际上反映出这些测度指标由于计算角度不同, 其本身也存在一定的差异. 通常最常用的测度指标是 RMSD.

除了提供最佳聚类中心, ESA 算法还给出了聚类参数的相关信息. 图 3 (a) 所示横坐标为 3 的位置处, $M_3 = 0$, 即 D_{ESA} 与 D_3 同时取得最优值. 实际上, ESA 算法所选择的结构正是 D_3. 因此, 可认为获得 D_3 的这次聚类所采用的参数正适合于图 3 (a) 所对应的相似性矩阵. 同样, 图 3 (a) 横坐标为 4 的位置处, $M_2 = 0$, 说明此处获得 D_2 的聚类参数适合于所对应的相似性矩阵. 而图 3 (a) 横坐标为 1 的位置处, 所有 $M_j < 0$, 说明此处 D_{ESA} 最优, 且 D_{ESA} 并非来源于 D_j $(j = 1, 2, 3, 4, 5)$, 这实际上与 D_j 的选择有关. D_j 来源于多次聚类尝试中较好的 5 次实验的最大 5 个类的聚类中心; 而 D_{ESA} 优选自所有 $N \times R_{\mathrm{MAX}}$ 个聚类中心, 即最佳聚类中心也许分布在其他多次聚类实验中.

3.3.2 ESA 算法结果与 SPICKER 结果的比较

在 I-TASSER 所产生的 6 个数据集 (1abv_, 1af7_, 1cqkA, 1csp_, 1dcjA, 256bA) 中, 同时提供了采用 SPICKER[3] 方法所产生的最优聚类中心. 本文还将 ESA 算法的结果与 SPICKER 所提供的结果进行比较.

由于 SPICKER 的实验数据所采用的测度指标是 RMSD, 因此本文在表 2 所示的实验数据中选择了测度指标为 RMSD 的 6 条数据, 在表 2 中的编号分别是 1, 4, 7, 10, 13 与 16. 此外, 2hq7_ 并非由 I-TASSER 提供, 也没有对应的 SPICKER 结果, 因此在这一组比较中并未采用 Data 19. 接下来, 用表 3 所示的 6 组实验中第 1 组、第 4 组实验的 ESA 算法结果 (这两组实验的评价指标也是 RMSD) 与 SPICKER 所提供的结果进行比较. 比较情况如表 5 所示.

表中第 2 列为所采用的数据集合 (见表 2); 第 3 和 4 列分别为实验 1 和 4 (见表 3) 的结果; 第 5 列为 SPICKER 提供的结果. 例如, 对于表 5 第 1 行数据, 在 1 号数据集合上的实验中, 第 1 次实验所选择的最佳聚类中心与天然结构之间的 RMSD 为

10.015, 第 2 次实验为 9.785, 而 SPICKER 所提供的聚类中心与天然结构之间的 RMSD 则为 13.941. 显然 ESA 的结果要优于 SPICKER.

表 5 与 SPICKER 的结果比较 (评价标准为 RMSD)
Table 5 Comparison with SPICKER (Structural metric is RMSD.)

No.	Data No.	Experiment 1	Experiment 4	SPICKER
1	Data 1	10.015	9.785	13.941
2	Data 4	7.897	8.878	4.728
3	Data 7	8.721	8.938	1.946
4	Data 10	2.427	2.427	2.369
5	Data 13	10.015	9.785	10.798
6	Data 16	3.438	3.227	3.448

综合表 5 的情况, 对于第 1, 5, 6 行数据, 实验 1 和 4 的结果均优于 SPICKER 的结果; 而第 2, 3, 4 行数据, 实验 1 和 4 的结果劣于 SPICKER. 总体情况二者相当. 由于 SPICKER 采用的蛋白质候选结构集合是第 3.1 节所描述的数据全集, 而本文实验的数据集是其十分之一子集. 因此, 表 5 的数据比较要依赖于该十分之一子集对数据集全集分布状态的表征能力. 尽管我们认为子集与全集具有近似的分布特征, 但这种比较实际上对 ESA 算法是不公平的. 然而, 即便是在这种劣势情况下, ESA 算法的结果仍然与 SPICKER 的结果相当.

4 结论

本文提出了一种对蛋白质结构聚类中心进行选择的算法. 由于蛋白质结构自由能分布的特性以及发现准确能量函数的困难性, 聚类成为蛋白质结构预测过程中必不可少的一个后处理步骤. 目前在蛋白质结构预测中常用的 QT 聚类算法依赖于由主观经验得出的聚类半径, 本文提出的聚类中心选择算法在缺乏足够经验数据的情况下, 通过对多次聚类结果进行分析, 从而选择最佳聚类中心, 进而确定聚类参数. 本文以 QT 算法中的聚类半径为出发点, 研究了 AP 算法中的 Preference 参数, 认为不同的聚类算法都存在一个影响聚类分布的特定参数. 本文提出的聚类中心选择算法同样适用于对 AP 算法的聚类结果进行分析. 实验结果表明, 算法能够在未知经验参数的情况下选择出最佳聚类中心, 同时也为不同聚类算法寻找适合相应数据的聚类参数提供了支持. 今后, 一方面将进一步研究蛋白质结构聚类中最佳聚类中心所对应的相关属性, 探索如何有指导地为多遍聚类过程设定不同的参数; 另一方面将探索如何引入更多的生物领域知识来指导聚类中心选择过程.

致谢

感谢美国南卡罗莱纳医科大学 (Medical University of South Carolina) 的金波博士为本文提出的修改意见.

References

1 Anfinsen C B. Principles that govern the folding of protein chains. *Science*, 1973, **181**(4096): 223−230

2 Bradley P, Misura K M S, Baker D. Toward high-resolution de novo structure prediction for small proteins. *Science*, 2005, **309**(5742): 1868−1871

3 Zhang Y, Skolnick J. SPICKER: a clustering approach to identify near-native protein folds. *Journal of Computational Chemistry*, 2004, **25**(6): 865−871

4 Wu S, Skolnich J, Zhang Y. Ab initio modeling of small proteins by iterative TASSER simulations. *BMC Biology*, 2007, **5**(1): 17−26

5 Zhang Y. Template-based modeling and free modeling by I-TASSER in CASP7. *Proteins: Structure, Function, and Bioinformatics*, 2007, **69**(S8): 108−117

6 Yue Feng, Sun Liang, Wang Kuan-Quan, Wang Yong-Ji, Zuo Wang-Meng. State-of-the-art of cluster analysis of gene expression data. *Acta Automatica Sinica*, 2008, **34**(2): 113−120
(岳峰, 孙亮, 王宽全, 王永吉, 左旺孟. 基因表达数据的聚类分析研究进展. 自动化学报, 2008, **34**(2): 113−120)

7 Moult J, Fidelis K, Kryshtafovych A, Rost B, Hubbard T, Tramontano A. Critical assessment of methods of protein structure prediction — round VII. *Proteins: Structure, Function, and Bioinformatics*, 2007, **69**(S8): 3−9

8 Heyer L J, Kruglyak S, Yooseph S. Exploring expression data: identification and analysis of coexpressed genes. *Genome Research*, 1999, **9**: 1106−1115

9 Wang Kai-Jun, Zhang Jun-Ying, Li Dan, Zhang Xin-Na, Guo Tao. Adaptive affinity propagation clustering. *Acta Automatica Sinica*, 2007, **33**(12): 1242−1245
(王开军, 张军英, 李丹, 张新娜, 郭涛. 自适应仿射传播聚类. 自动化学报, 2007, **33**(12): 1242−1245)

10 Frey B J, Dueck D. Clustering by passing messages between data points. *Science*, 2007, **315**(5814): 972−976

11 Shortle D, Simons K T, Baker D. Clustering of low-energy conformations near the native structures of small proteins. *Proceedings of the National Academy of Sciences of the USA*, 1998, **95**(19): 11158−11162

12 Xiao Yu, Yu Jian. Semi-supervised clustering based on affinity propagation algorithm. *Journal of Software*, 2008, **19**(11): 2803−2813
(肖宇, 于剑. 基于近邻传播算法的半监督聚类. 软件学报, 2008, **19**(11): 2803−2813)

13 Liu Ming, Wang Xiao-Long, Liu Yuan-Chao. A fast clustering algorithm for large-scale and high dimensional data. *Acta Automatica Sinica*, 2009, **35**(7): 859−866
(刘铭, 王晓龙, 刘远超. 一种大规模高维数据快速聚类算法. 自动化学报, 2009, **35**(7): 859−866)

14 Ni Wei-Wei, Sun Zhi-Hui, Lu Jie-Ping. K-LDCHD — a local density based k-neighborhood clustering algorithm for high dimensional space. *Journal of Computer Research and Development*, 2005, **42**(5): 784−791
(倪巍伟, 孙志挥, 陆介平. K-LDCHD — 高维空间 k 邻域局部密度聚类算法. 计算机研究与发展, 2005, **42**(5): 784−791)

15 Hubert M, Veeken S V. Outlier detection for skewed data. *Journal of Chemometrics*, 2008, **22**(3−4): 235−246

16 Zhang Y. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics*, 2008, **9**(1): 40−47

17 Rohl C A, Strrauss C E M, Misura K M S, Baker D. Protein structure prediction using Rosetta. *Methods in Enzymology*, 2004, **383**: 66−93

18 Kryshtafovych A, Milostan M, Szajkowski L, Daniluk P, Fidelis K. Casp6 data processing and automatic evaluation at the protein structure prediction center. *Proteins: Structure, Function, and Bioinformatics*, 2005, **61**(S7): 19−23

19 Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 2004, **57**(4): 702−710

20 Tress M, Ezkurdia I, Grana O, Lopez G, Valencia A. Assessment of predictions submitted for the CASP6 comparative modeling category. *Proteins: Structure, Function, and Bioinformatics*, 2005, **61**(S7): 27−45

黄 旭 苏州大学计算机科学与技术学院博士研究生. 主要研究方向为生物信息计算, 元启发搜索和并行分布计算.
E-mail: huangxu_sd@163.com
(**HUANG Xu** Ph. D. candidate at the School of Computer Science and Technology, Soochow University. His research interest covers calculation of biological information, meta heuristics search, and parallel and distributed computing.)

吕 强 苏州大学计算机科学与技术学院教授. 主要研究方向为生物信息计算, 元启发搜索和并行分布计算. 本文通信作者. E-mail: qiang@suda.edu.cn
(**LV Qiang** Professor at the School of Computer Science and Technology, Soochow University. His research interest covers calculation of biological information, meta heuristics search, and parallel and distributed computing. Corresponding author of this paper.)

钱培德 苏州大学计算机科学与技术学院教授. 主要研究方向为中文信息处理技术, 分布计算和操作系统.
E-mail: pdqian@suda.edu.cn
(**QIAN Pei-De** Professor at the School of Computer Science and Technology, Soochow University. His research interest covers Chinese information processing, distributed computing, and operating system.)