软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(8):1827−1837 [doi: 10.3724/SP.J.1001.2011.03848]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种有效的蛋白质序列聚类分析方法*

唐东明, 朱清新+, 杨 凡, 陈 科

(电子科技大学 计算机科学与工程学院,四川 成都 610054)

Efficient Cluster Analysis Method for Protein Sequences

TANG Dong-Ming, ZHU Qing-Xin+, YANG Fan, CHEN Ke

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

+ Corresponding author: E-mail: qxzhu@uestc.edu.cn

Tang DM, Zhu QX, Yang F, Chen K. Efficient cluster analysis method for protein sequences. *Journal of Software*, 2011,22(8):1827−1837. http://www.jos.org.cn/1000-9825/3848.htm

Abstract: This paper proposes an efficient clustering method for protein sequences, using Affinity propagation algorithm (AP) and post-processing. In order to optimize the clustering result, post-processing is used to improve the clustering result of AP. To measure the similarity between two protein sequences, an improved alignment-free similarity measure is presented. This method is evaluated and compared with other algorithms on six protein sequences data sets. Experimental results demonstrate the effective performance of the proposed method.
Key words: pattern recognition; cluster analysis; sequence analysis; protein sequence

摘 要: 提出了一种有效的基于仿射传播聚类算法和后处理方法的蛋白质序列聚类方法.在聚类分析蛋白质序列时,为了优化仿射传播聚类算法的聚类结果,采用后处理的方式来提高聚类结果的质量.为了度量蛋白质序列之间的相似度,给出了一种改进的无比对计算方法.在6个蛋白质序列数据集上进行对比实验,实验结果表明,所给出的方法能够有效地分析蛋白质序列.
关键词: 模式识别;聚类分析;序列分析;蛋白质序列
中图法分类号: TP391 **文献标识码**: A

聚类算法作为一种有效的数据分析方法被广泛应用于数据挖掘、模式识别、机器学习、图像分割、语音识别、生物信息处理等领域.聚类算法还可以应用于商业分析.它可以帮助市场决策人员从消费者数据库中区分出不同的消费群体,并且概括出每一类消费者的消费模式或习惯.本质上说,聚类算法是将总体中的个体分类以发现数据中的结构,希望一个类中的个体彼此接近或相似,而与其他类中的个体相异.这样就可以对划分出来的每一类进行深入的分析,从而概括出每一类的特点.文献[1,2]对各种聚类算法进行了详细的描述.

目前,生物信息学研究已经是生命科学领域和计算机科学领域相交叉的一个研究热点.生物信息学研究成果已经被广泛应用于基因发现和预测、基因数据的存储管理、数据检索与挖掘、基因表达数据分析、蛋白质结构预测、基因和蛋白质同源关系预测、序列分析与比对等.蛋白质是组成生物体的基本物质,是生命活动的主要承担者.一切生命活动无不与蛋白质有关.通过多年的积累,目前已经形成了海量的蛋白质序列数据库.面

* 基金项目: 国家自然科学基金(60671033)
收稿时间: 2008-12-01; 修改时间: 2009-07-29; 定稿时间: 2010-03-04

对庞大的序列数据库,如果仅仅根据序列信息就可以确定蛋白质的家族信息,那么将有助于分析相似蛋白质的功能差异性,并且揭示蛋白质之间的相互作用原理.目前,实验测出的蛋白质序列数量已经远远超出已经确定功能的蛋白质序列数量.如何根据现有的已经确定功能的蛋白质序列来分析预测新的蛋白质序列的功能并鉴别蛋白质序列之间的差异性,是摆在生物学家面前的重要问题.聚类分析通过测量蛋白质序列之间的相似性,对蛋白质序列进行有效的划分,为确定蛋白质序列的家族信息和预测蛋白质序列的功能及对蛋白质序列进行同源检测提供了有力的依据.在实际应用中一般将未确定功能的蛋白质序列和已知功能的蛋白质序列混合进行聚类分析,通过已定义的功能类信息推测未知功能的蛋白质序列功能信息.文献[3]对聚类算法在生物信息学领域的应用与研究进展给出了详细的综述.

目前已经有很多聚类方法用于蛋白质序列分析.从算法采用的聚类策略来说,它们主要分为两类[4]:(1) 基于完全连通图的方法,在图中顶点表示蛋白质,赋权的边表示蛋白质之间的距离,通过对图中的边进行裁剪操作获得一个特定阈值下的分类结果,如 GeneRAGE[5]和 FORCE[6];(2) 使用单连接聚类组织构建一个层次树以反映蛋白质序列之间的距离,使用一个阈值从层次树获得分类结果,如 ProtoMap[7],SYSTERS[8]和 ProClust[9].

从蛋白质序列聚类算法采用的相似度计算方式来说,它们又可以分为有比对和无比对算法两类.前者在计算序列之间的相似度时依赖序列比对软件的结果,序列比对软件一般采用 BLAST[10];后者在计算相似度时并不依赖序列比对软件的结果,而是直接依据序列本身的信息来计算序列之间的相似度.GeneRAGE,FORCE,ProtoMap,SYSTERS,ProClust,Spectral[4]和 TribeMCL[11]均属于有比对的蛋白质序列聚类算法.在实际应用中发现,对一些蛋白质序列数据集进行全序列比对有时并不能给出结果,尤其是对含有难比对的蛋白质序列的数据集,一个长序列和一个短序列有时也无法给出两者之间的相似度得分.在这种情况下,有比对的蛋白质序列聚类算法就无法进行正确的聚类分析.近年来,无比对的聚类算法受到了关注.CLUSS[12]提出了新的蛋白质序列相似度计算方法 SMS(substitution matching similarity),并通过层次化聚类方法生成系统进化树,对进化树节点赋予协相似度值,最后通过阈值来划分不同的蛋白质家族.无比对蛋白质序列聚类算法的一个核心任务就是构建无比对的序列相似度计算方法.文献[13]对早期的无比对成对序列比较的方法给出了一个详细的综述.

虽然目前已经产生了大量优秀的用于蛋白质序列分析的聚类算法,但是由于蛋白质序列数量庞大,且目前已经确定功能的蛋白质序列并不是很多,因此,目前用于蛋白质序列聚类的方法都存在一些局限性,面临着以下几个问题和挑战:(1) 如何精确地计算序列之间的相似度;(2) 减少算法对参数设定的依赖,如聚类个数设定、阈值设定等;(3) 如何在聚类精度和时间效率之间进行平衡;(4) 如何评价产生的聚类结果;(5) 易用性.

2007 年,Frey 等人在《Science》上提出了一种聚类算法——仿射传播聚类算法(affinity propagation clustering,简称 AP)[14].AP 算法能够快速地处理大规模数据的聚类问题,并且在很多应用中取得了较好的结果,但是在对蛋白质序列进行聚类分析时,它目前还面临着一些问题.后面本文将给出问题的详细描述.为了解决这些问题,本文提出了一种有效的基于仿射传播聚类算法和后处理方法的蛋白质序列聚类方法(post-process affinity propagation clustering,简称 ppAP).

本文主要包含以下几个部分:(1) 简要介绍 AP 算法,并分析其在实际应用中出现的问题;(2) 给出一种改进的无比对的蛋白质序列之间相似度计算的方法;(3) 详细描述所提出的方法;(4) 实验验证,并与其他方法进行对比.

1 仿射传播聚类

设给定的 N 个数据点为 $x_1,x_2,\ldots,x_N$,为了划分这 N 个数据点为不同的类别,AP 引入了“类代表(exemplar)”的概念,即一类数据中的代表点,算法在初始时将所有数据点都看作潜在的类代表点.AP 将每个数据点看作网络中的一个节点,通过节点之间不断迭代传送更新的真实数据消息来产生类代表点及相应的附属点,在消息传递的过程中就完成了数据的分簇.相似度矩阵 s 中的 $s(i,k)$表示数据点 k 适合作为数据点 i 的类代表点的度.在消息传递开始前,每个数据都被当作潜在的类代表点,在消息传递过程中有两类消息,数据点 k 为了成为类代表点,需要从每个数据点 i 收集证据 $r(i,k)$,它表示数据点 k 对点 i 的“responsibility”,用来度量数据点 k 适合作为数据点 i

的类代表的程度,即点 k 对点 i 的吸引度;在消息传递过程中,$r(i,k)$的更新方式如公式(1)所示.数据点 i 从候选的类代表点 k 收集证据 $a(i,k)$,表示数据点 i 对点 k 的“availability”,用来度量数据点 i 选择点 k 作为它的类代表点的适合程度,即点 i 对点 k 的归属度;在消息传递过程中,$a(i,k)$的更新方式如公式(2)所示.每一次迭代都包含 3 个部分:(1) 更新吸引度以给出适应度;(2) 更新所有点的适应度以给出吸引度;(3) 给出吸引度和适应度监控类代表的决策,并当 n 次迭代都没有变化的情况下终止算法.

AP 和传统的 k-means 聚类算法不同.它不需要预先知晓聚类的个数,通过一个称为 preferences 即偏度 p 的参数来调节聚类的个数.该参数可以为单一的数值,也可以是一个 $1\times N$ 的向量.单一值则表示每个数据点的偏度都为该值,向量则表示每个数据点的偏度对应向量中相应位置的元素值.可以为每个数据点指定一个偏度值,值越大,则数据点越可能被选为类代表点.如果预先并不知晓哪些数据点适合作为类代表点,则可以为所有数据点指定一个相同的值.在消息传递开始时,$s(k,k)$被赋值为数据点 k 的偏度值.

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

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

AP 不需要提前给出聚类的个数,从消息的传递过程中并且依赖输入的偏度来产生适当的聚类数,这样就能基于一个先决适当条件进行自动的模型选择,通过调节偏度就能调节节点作为类代表的度.AP 可以处理相似度非对称($s(i,k)\neq s(k,i)$)且不满足三角不等式($s(i,k)<s(i,j)+s(j,k)$)的情况.AP 可以看作是一种搜索最小能量函数的方法,用 $c_1,c_2,\ldots,c_N$ 表示相应的 N 个数据点的类代表,则 $s(i,c_i)$就是数据点 i 到它的类代表点的相似度.因此,相应的能量函数为 $E(C)=-\sum_{i=1}^{N}s(i,c_i)$. 实际上,计算最小能量是很难的,是一个 NP 难度 k 中值问题.但是,AP 的更新迭代规则协调了固定点的递归来最小化为一个近似 Bethe 自由能量.

AP 算法已经被应用到很多问题的研究中,如,文献[15]采用 AP 来鉴别复杂网络的群组;文献[16]将线性判别分析和 AP 结合来进行人面识别;文献[17]使用已知的标签数据或者成对点约束对数据形成的相似度矩阵进行调整,进而提出了一种半监督的 AP 算法.但在实际使用中,AP 算法面临着一些问题.首先,从前面的叙述可知,偏度 p 的大小对聚类的结果影响很大.文献[14]推荐,一般情况下采用相似度矩阵的中值来作为所有数据点的共同偏度值,但是在很多情况下,使用中值并不能给出较好的聚类结果.图 1 中给出了采用文献[14]给出的北美航空旅行数据在不同的偏度值下获得的聚类结果.从图中可知,p 值对聚类的结果影响非常大.因此在实际使用 AP 时,如何选择合适的偏度以得到较好的聚类结果,是一个需要解决的问题.其次,AP 采用单一的类代表点来表示类,显然就像 k-means 算法采用中心点来表示类一样,它不能对非一致分布的数据进行较好的划分,例如环状数据分布.由于上述原因,AP 的聚类结果可能将数据集中的大类划分为很多个小类,并且将一些小类错误地合并.

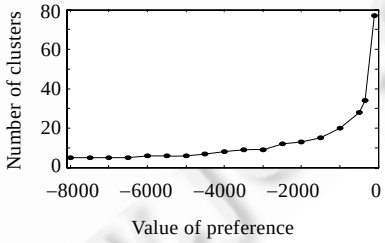


Fig.1 Effect of the preference on the result of clustering

图 1 偏度对聚类结果的影响

通过给每个数据点赋予不同的偏度值,可能可以部分地解决上面描述的问题.但是在大部分应用场合,很难知道数据集的先验分布知识,因此很难给每个数据点赋予合适的偏度值,只能给所有数据点赋予相同的偏度值.目前,已经有学者提出一些方法来改进 AP 算法,文献[18]给出了一种自适应的仿射聚类算法(adAP),其基本原理是,从一个大的$|p|$值开始,以一定的步幅下降,自适应地扫描偏度参数空间来搜索聚类个数空间,以寻找最优聚类

结果.由图 1 可以看出,p 值小到一定程度后,聚类的结果趋于稳定,因此,调节 p 值能够取得的最好聚类结果是受 AP 自身限制的.本文提出的方法绕过偏度赋值,采用结合后处理的方式来提升聚类的结果.

2 蛋白质序列相似度计算

蛋白质序列之间的相似度计算是蛋白质序列聚类分析中的一个重要步骤.相似度度量的准确性直接关系到聚类结果的好坏.目前,已知的相似度计算方法可以分为两类:(1) 有序列比对软件参与的;(2) 无序列比对软件参与的,即直接基于序列信息进行计算.有序列比对的相似度计算方法一般采用序列比对软件的结果来计算相似度,如采用 BLAST[10]及其改进方法的 E 值作为距离度量.一般最常用的计算方法为 Best hit(BeH),即 E 值的负对数最大值,FORCE[6],Spectral[4]和 TribeMCL[11]即采用 BeH 方法.另外还有一些方法,如文献[19]采用的基于片段转移的方法.

有比对的计算方法的一个假设前提就是序列中的同源片断是邻接的,但是这个假设在分子序列中有时并不可靠,如可变剪切等.因此,这些有比对的计算方法目前存在着一些问题,难以得到较高准确性的结果,尤其是对于一些难以比对的序列,如多域、循环交换(circular permutation)和 tandem 重复的蛋白质序列.并且,有比对的计算方法在使用中比较繁琐,需要首先采用 BLAST 等软件对序列集合进行 all-vs-all 的序列比对,然后解析结果文件,再计算相似度,因此难以用于大规模的聚类分析.目前,无比对的计算方法受到了广泛的关注.

文献[12]提出使用 SMS(substitution matching similarity)相似度度量来计算蛋白质序列相似度.该方法首先找出两个序列之间长度超过一个阈值的所有确切匹配的子序列,然后基于这些子序列的得分计算出相似度.SMS 特别适用于难以比对的序列,如多个结构域的蛋白质.但是,SMS 也存在一些问题:首先,SMS 只考虑了两个序列之间相同的子序列对;其次,它没有考虑不匹配的序列.对于相似度高的序列,考虑相同的子序列就足够了;但是对于相似度低的序列,它们之间完全匹配的子序列是很少的,如果只考虑这些子序列将会对相似度的计算产生较大的偏差.基于上面的分析,下面给出一种改进的相似度度量的方法.该方法的中心思想是,将限制条件放宽,尽可能多地考虑两个序列之间所有相似的子序列对.只要子序列之间的相似得分大于一个阈值,就认为它们之间相似,然后再考虑去掉匹配的子序列后的序列的相似度,再将两者相加.

长度为 n 的蛋白质序列可以看成是一个有 n 个符号的线性序列,每个符号可取值为任意一种残基.使用下标来索引序列中的残基:X_i 表示序列 X 中的第 i 个残基,$X_{i...j}$ 表示序列 X 中的第 i 个到第 j 个残基的子序列,$|X|$ 表示序列 X 的长度.对于一个区间$[i,j]$,如果满足 $i,j<|X|$,则 $X_{i...j}$ 为序列 X 的子序列.因此,如果$[i,j]\subset[i',j']$,则意味着子序列 $X_{i'...j'}$覆盖 $X_{i...j}$,反之亦然.设 X 和 Y 为两个要计算两者之间相似度的蛋白质序列,x 和 y 分别是 X 和 Y 的子序列,用 $p_{x,y}$ 表示 x 和 y 形成的序列对,设$|p_{x,y}|$为 $p_{x,y}$ 的长度,则有$|p_{x,y}|=\max\{|x|,|y|\}$.为了区分每个子序列对的重要性,给每个子序列对定义一个分值 $W(p_{x,y})=\sum_{i=1}^{|p_{x,y}|}M(x[i],y[i])$.该值表示子序列 x 和 y 之间的最佳得分,用来评估子序列 x 和 y 之间的相似度,也可以当成是对这对蛋白质片断的保守性估计.M 为记分矩阵,在此可以选用 PAM62,BLOSUM250 中的一种,在实际应用中它们区别不大;$x[i],y[i]$分别表示匹配的子序列对 x 和 y 的第 i 个残基.对两个子序列对 $p_{x,y}$ 和 $p_{x',y'}$,且 $x=X_{i...j},y=Y_{u...v},x'=X_{i'...j'},y'=Y_{u'...v'}$,如果$[i,j]\subset[i',j']$,$[u,v]\subset[u',v']$或者$[i',j']\subset[i,j]$,$[u',v']\subset[u,v]$,则认为 $p_{x,y}=p_{x',y'}$.定义所有匹配的子序列对 $p_{x,y}$ 的集合为 $C=\{p_{x,y}\}$,$|p_{x,y}|\geqslant l$,$W(p_{x,y})\geqslant e$,依照前面所述,如果较长的匹配的子序列对包含较短的匹配的子序列对,则集合 C 中只计入较长的匹配的子序列对.从集合 C 的定义可以看出,匹配的子序列对只考虑了长度大于 l、相似度得分大于 e 的子序列对.两个序列之间所有匹配的子序列对的相似度得分则为 $SSim(X,Y)=\sum_{p\in C}W(p)/\max(|X|,|Y|)$.

下面计算除去匹配的子序列后的序列的相似度,对于这些序列,基于比对的度量和基于无比对的度量有着相同的区分能力.因此,可以采用无比对的计算量小的标准 Euclidean 来度量相似度低的序列.标准 Euclidean 是基于 L-tuple 的,一个 L-tuple 是 L 个符号的片断.设集合 Z_L 包含序列中所有可能的 L-tuple,且其长度为 K,则集合 Z_L 的定义为 $Z_L=\{Z_{L,1},Z_{L,2},\ldots,Z_{L,K}\}$,$K=20^L$.使用序列 X 的 L-tuple 计数值 c_L^X 将 X 映射到 Euclidean 空间中的一个

向量 $c_L^X=(c_{L,1}^X,...,c_{L,k}^X)$,式中,$c_{L,i}^X$ 是序列 X 中 L-tuple $z_{L,i}$ 考虑重叠时出现的次数.使用标准 Euclidean 距离来表示剩下的序列的相异度,设序列 $\hat{X}$ 和 $\hat{Y}$ 是除掉匹配的子序列后的序列,那么标准 Euclidean 距离可以定义为 $d^{SE}(\hat{X},\hat{Y})=(c^{\hat{X}}-c^{\hat{Y}})^T\cdot[diag(s_{11},...,s_{KK})]^{-1}\cdot(c^{\hat{X}}-c^{\hat{Y}})$,式中,$s_{11},\ldots,s_{KK}$ 为序列的 L-tuple 计数向量协方差矩阵的对角元素.为了将距离转换成相似度,可以使用负指数函数的方式,因此最终的序列之间的相似度为 $Sim(X,Y)=SSim(X,Y)+f(d)$,其中,$f(d)$为负指数函数.

3 聚类分析方法

3.1 聚类方法过程描述

由前面的叙述可知,AP 采用相似度矩阵的中值得到的聚类结果并不是最优的结果.为了得到最优的蛋白质聚类结果,受 FORCE[6]的启发,本文提出了一种结合 AP 算法和后处理方法的蛋白质序列聚类方法(ppAP).ppAp 主要包括 3 个阶段:(1) 根据序列信息来计算序列之间的相似度;(2) 使用相似度矩阵的中值作为共同偏度值来调用 AP 进行初步的聚类分析;(3) 对 AP 的聚类结果进行后处理.其中,后处理阶段包含以下两步:

(1) 合并:ppAP 的第 2 阶段使用中值作为共同偏度值调用 AP 对蛋白质序列集进行聚类分析,尽管没有产生最优的聚类结果,但是也对数据进行了初步的划分.为了整合分散孤立的个体类以使聚类结果变得紧密,对得到的聚类结果进行合并操作.首先对得到的聚类结果按类成员的个数进行升序排序,设所得的结果为 $sortOfCluster=\{c_1,c_2,...,c_T\}$,从 c_1 到 c_T 依次尝试与其他类进行两两合并,直到找到一个优于 $sortOfCluster$ 的聚类结果 $mergeOfCluster=\{c_1,c_2,...,c_i\cup c_j,c_T\}$,对 $mergeOfCluster$ 进行排序以进行新的合并操作,直到没有新的合并操作为止.为了评价每次合并尝试的结果是否优于合并前,本文采用 Silhouettes[20]值来评判.Silhouette 值可以用来评价聚类的类内结构的紧密性和可分性,可以用于评价聚类的结果.

设具有 n 个样本点的数据集聚类划分的结果为 $C=\{c_1,c_2,\ldots,c_L\}$,样本点 x_i 被划分到 c_j 类中,则样本点 x_i 的 Silhouettes 指标 $sil(i)(i=1,\ldots,n)$表示样本点 x_i 被划分到 c_j 类中的质量,$sil(i)$按公式(3)定义:

$$sil(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{3}$$

公式(3)中的 $a(i)$表示样本点 x_i 与 c_j 类中其他样本点的平均不相似度或距离,$b(i)$表示样本点与除 c_j 类之外的其他类的平均不相似度或距离的最小值.从公式(3)可以看出,$sil(i)$的取值范围为−1~1.当值接近 1 时,表明样本点被划分到一个恰当的类;值接近 0 时,表明样本点被划分到了一个最近邻类,比如样本点和其他类的距离基本相同的情况;当值接近−1 时,表明样本点被划分到了一个不恰当的类中.由 $sil(i)$可以计算出聚类结果的平均 Silhouettes 值 $silAV=\frac{1}{n}\sum_{i=1}^{n}sil(i)$.该值可以反映聚类结果的质量.每次合并尝试后,通过该值来评判本次尝试是否有利于获得更优的聚类结果:如果有利,则保留本次尝试;否则,抛弃并进行下一次尝试.

(2) 调整:对上一步获得的聚类结果依照类成员的个数按升序进行排序,设所得的结果为 $sortOfCluster=\{c_1,c_2,...,c_M\}$,从 c_1 到 c_M 依次尝试与其他类进行调整操作,即将类 c_i 中的成员 x_k 依次尝试移动到类 c_j 中,直到找到一个优于 $sortOfCluster$ 的聚类结果 $adjOfCluster$.对 $adjOfCluster$ 进行排序以进行新的调整操作,直到没有新的调整操作为止.为了评价每次调整尝试的结果是否优于调整前,同样采用 Silhouettes 值来评判.

3.2 方法分析

下面结合实验部分给出的 COG_765 蛋白质序列数据集来分析 ppAP 方法.ppAP 在第 1 阶段,直接读取以 FASTA 格式存储的序列文件,并计算序列之间的相似度.图 2 给出了 ppAP 的第 2 阶段 AP 给出的聚类结果图中两条横的虚线之间表示算法获得的一个分类,竖的长实线用来分隔数据集中“真实”的类,两条虚线之间出现的短竖线表示一条蛋白质序列.最终一个蛋白质序列出现在哪个位置要由它本身属于哪个类和算法最终将它划分到哪个类来共同决定.从图 2 中不难发现,AP 对数据集给出了一个初步的划分,如对一些类成员数量比较小的类划分较为正确.但与正确的类分布相比,AP 给出的聚类结果产生了过多的类,并且 AP 将数据集中的大类划分

成了很多分散的小类.同时可以看出,AP 的聚类结果倾向于平均划分数据集,即聚类结果的每个类成员数量个数接近,这个特性类似于 *k*-means 方法.在此,我们引入变异系数 CV(coefficient of variation)[21]来评估聚类划分后的类成员个数分布和数据集本身"真实"的类成员个数分布之间的差异.设 L 是由聚类划分结果的每个类的成员个数组成的集合,则集合 L 的变异系数 $CV=\sigma/\mu$,σ表示集合的方差,μ表示均值.变异系数 CV 可以度量集合中的数据分布的变化,当 CV 值大于 1 时,表示集合中的数据变化比较大;反之,则表示变化比较小.数据集 COG_765 本身"真实"的 CV 值为 1.58,ppAP 的第 2 阶段,AP 给出的聚类结果的 CV 值为 0.83.显然,AP 的聚类结果倾向于均匀化分类.

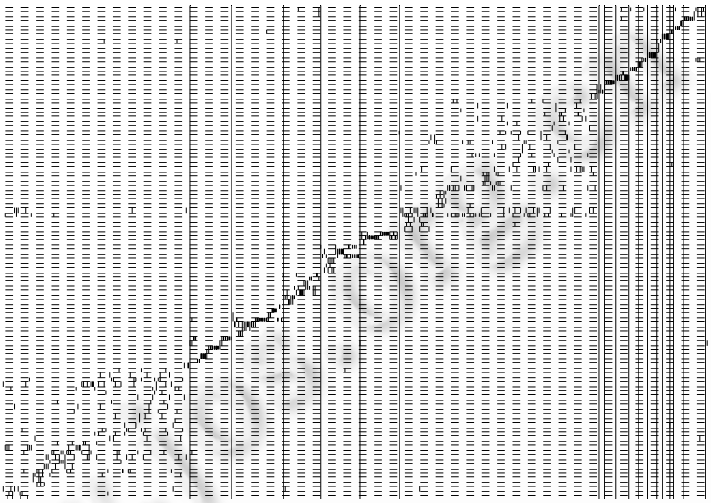


Fig.2 Clustering result of the second stage of ppAP on the data set COG_765

图 2 ppAP 的第 2 阶段在 COG_765 数据集上获得的聚类结果

由第 2 节关于 AP 的描述可知,AP 可以看作是一种搜索最小能量函数 $E(C)=-\sum_{i=1}^{N}s(i,c_i)$ 的方法.在这个搜索最小能量的过程中,采用类代表的方式来表示一个类的虚拟"中心".如果数据集中类的成员和其他类的成员距离或者相似度差距变化并不明显,此时,一个大类中就很难找出一个"强势"的类代表点来表示这个类,这样就会形成很多局部的"类代表"点来代表局部的成员.由于生命遗传变异的复杂性,蛋白质序列的可分性并不是非常明显,因此目前用于蛋白质序列分析的无监督聚类算法都面临的一个共同的难题,就是现有的测量序列之间的相似度的方法产生的数值结果直接可分性都不高.由于 BLAST 对一些序列无法进行比对,尤其是对难比对的序列,因此在使用 BLAST 进行序列比对时可能只能探测到部分的显著匹配序列,这样就会导致在计算相似度时数据集中的大部分序列之间无法计算出相似度.在这种情况下,基于 BLAST 比对结果的蛋白质序列聚类算法的聚类结果就会产生大量的"孤类",即只有 1 个成员的类.TribeMCL[11]采用 BLAST 结果的 E 值的负对数作为蛋白质序列之间的相似度.在实验中使用 TribeMCL 进行蛋白质序列分析时就会发现,聚类的结果包含大量的"孤类".从搜索最小能量函数的角度来看,图 2 给出的聚类结果可能进入了一种局部最优的情况.聚类算法从本质上来讲都有一个目标函数,聚类的过程就是一种寻找最小化或最大化目标函数的过程.聚类方法中的寻找最大化或最小化目标函数的问题一般都可以归结为 NP 难问题.求解这类问题的过程一般都是通过多次的迭代尝试求解,因此容易陷入局部最优.为了防止陷入局部最优,聚类的过程中一般采用设置阈值和限制尝试次数的方法.从理论上很难对出现的这些情况给出一个深入详细的分析,因为这之间的关系比较复杂且受很多因素影响,如数据本身的分布、序列之间表现出的亲缘性等.

结合后处理的方法可以有效地改善聚类算法的性能,如 ProClust[9]的第 1 阶段采用 Bolten 等人[22]给出的方法对蛋白质序列数据集进行初步划分,再使用 Profile-HMMs 的方法来评估各个类并进行相应的合并操作;Li[23]

使用后处理的方式来合并聚类中的一些小类;FORCE[6]采用后处理的方式来调整第 1 步采用求解加权图编辑问题获得的蛋白质序列分类结果.

从前面的描述可知,整个后处理阶段就是一个寻找最大化目标函数 $silAV=\frac{1}{n}\sum_{i=1}^{n}sil(i)$ 的过程.从对公式(3)的描述可知,最大化目标函数 silAV 的过程即寻求类划分结果类内成员之间关系紧密而类间关系疏远的过程.公式(3)中的 $a(i)$即表示类成员和类内其他成员之间的关联程度,$b(i)$即表示类成员与其他类的成员的关联程度.

后处理的合并过程将前一阶段 AP 形成的分散的较小的类合并.由于第 1 阶段 AP 获得的划分中有些不同家族的蛋白质序列被错误地划分到了一个类中,并且由于合并的过程中也可能存在局部最优合并的情况,这种情况可以理解为在迭代过程中某一次合并 silAV 值虽然大于合并前,但是可能对后续的迭代合并产生不良影响.为了处理这种情况,有些算法采用设置阈值的方式来逃离.但是在实际使用中,这个阈值并不好确定.因此,本文采用一个调整过程来对第 1 步合并的结果进行调整.平均 Silhouettes 值函数是一个聚类结果的无先验知识评价函数.在聚类的过程中结合聚类结果评估函数来调整聚类的运行过程,在目前已知的聚类算法中并不鲜见,如 Tseng 等人[24]给出的用于基因表达数据聚类的 CST 算法.CST 算法可以看成是一个通过合并过程和移除过程来寻求最大化Γ统计检验函数的过程.寻求这类聚类校验函数最大化的过程本身就是一个寻求最好聚类划分的过程,因此本文给出的后处理过程本质上与聚类的目标是一致的.本文第 4 节的图 4(c)给出了 ppAP 最后给出的聚类结果,从图中可以看出,经过后处理,聚类结果变得较为紧密,此前形成的大量的小类被合并到了一起,并且聚类结果较为接近“真实”的划分.最后结果的 CV 值为 1.38,也较为接近数据集“真实”的类分布.

4 实验结果及分析

4.1 材料

实验用的蛋白质序列数据集来源于蛋白质结构分类数据库 SCOP(structural classification of proteins)[25,26]和蛋白质直系同源簇 COG(clusters of orthologous groups of proteins)[27]数据库.蛋白质结构分类数据库 SCOP 由英国医学研究委员会(medical research council,简称 MRC)的分子生物学实验室和蛋白质工程研究中心开发和维护.SCOP 的目标是提供关于已知结构蛋白质之间的结构和进化关系的信息.该数据库对已知三维结构的蛋白质进行分类,并描述了它们之间的结构和进化关系,所涉及的蛋白质包括结构数据库 PDB 中的所有条目.

蛋白质直系同源簇(COG)数据库是对细菌、藻类和真核生物的 21 个完整基因组的编码蛋白,根据系统进化关系分类构建而成.COG 库对于预测单个蛋白质的功能和整个新基因组中蛋白质的功能都很有用.本文采用的 SCOP 蛋白质序列数据为随机从 SCOP 数据库中获得,COG 数据集采用和文献[12]相同的数据集,对 SCOP 数据,我们采用超家族信息作为数据集“真实”的分类结果;对 COG 数据集,我们采用其家族信息作为“真实”的分类结果.表 1 给出了实验用数据集的有关信息.本文采用的数据集均比较“杂乱”,每个类的成员个数并不平均,有些类的成员个数非常多,有些个数非常少,甚至有一些只有 1 个或两个成员的“孤类”.文献[12]给出的 COG 数据集均含有部分重复的序列,为了验证算法在苛刻条件下的性能,本文在实验中并没有将重复的序列去除.

Table 1 Datasets used in experiment

表 1 实验中使用的数据集

	SCOP_200	SCOP_512	COG_423	COG_487	COG_584	COG_765
Database	SCOP	SCOP	COG	COG	COG	COG
Number of sequences	200	512	423	487	584	765
Number of clusters	7	20	20	19	19	19

4.2 实验内容和结果分析

为了评估实验的结果,本文采用 F-measure[4]来评估聚类的结果.对一个给定的蛋白质序列集,设 $K=\{k_1,k_2,...,k_m\}$为实验中获得的聚类结果,且 $C=\{c_1,c_2,\ldots,c_T\}$为蛋白质序列集的正确分类,N 为蛋白质序列集中的序列总数,

n_i 和 n^j 分别为类 k_i 和 c_j 的序列个数,n_i^j 为 k_i 和 c_j 的交集 $k_i \cap c_j$ 中的序列个数,则 F-measure 如公式(4)所示.F-measure 的取值为 0,1 之间,值越大,说明聚类的结果越好,值为 1 则表明聚类结果和正确的分类完全相同.

$$F(K,C)=\frac{1}{N}\sum_{j=1}^{T}n^j\times\max_{1\leqslant i\leqslant m}\left(\frac{2n_i^j}{n_i+n^j}\right) \tag{4}$$

为了测试本文提出的 ppAP 方法的性能,我们将 ppAP 分别与谱聚类算法[4],TribeMCL[11],CLUSS[12],BlastClust[10],AP[14]及 adAP[18]算法进行对比实验.文献[4]给出的谱聚类算法是将 Ng,Jordan 和 Weiss algorithm (NJW)[28]谱聚类算法改进后用于蛋白质序列聚类,并且使用 BLAST 结果作为序列之间的相似度;TribeMCL[11]算法采用 BLAST 结果作为序列之间的相似度并构建相似度矩阵,并且构建一个加权的转移概率图,最后通过一种模拟的随机行走来对蛋白质序列进行划分;CLUSS[12]是一个无比对的层次化聚类算法;BlastClust[10]算法是 BLAST 软件包提供的一个单连接层次聚类算法,同样采用 BLAST 的结果作为相似度.

实验中,AP,adAP 和 ppAP 均采用相同的相似度矩阵且均采用默认参数.该矩阵由本文第 2 节提出的计算方法得出.其中,AP 算法的偏度 p 值取为相似度的中值,其他参数为默认值.TribeMCL,BlastClust 和谱聚类算法的 E-value cut-off 均设置为 0.001,其他参数均采用默认参数.采用默认参数是符合蛋白质序列聚类算法实际使用环境的,因为生命科学人员采用聚类算法来辅助分析序列时,大部分情况是没有太多的先验知识来调节设置参数的,通常情况下都是采用默认配置来调用聚类算法.

实验获得的每种算法聚类结果的 F-measure 值统计结果见表 2.从表 2 中可以看出,谱聚类算法和 ppAP 表现较为稳定,对 6 个数据集均取得了较好的聚类效果.从表中可以看出,CLUSS 对一些数据集取得了较好的聚类结果,尤其是对 COG_423 数据集取得最高的 F-measure 值 0.873 9,并且对 COG_487 数据集也取得了不错的聚类结果.但是 CLUSS 算法似乎对数据集较为敏感,表现起伏较大,不能稳定地获得较好的聚类结果,对其他几个数据集的聚类效果并不太理想.TribeMCL 和 BlastClust 的聚类结果产生了大量的“孤类”,因此聚类结果并不理想.这个结果与本文第 3.2 节的分析一致.对比表中的 AP,adAP,ppAP 聚类结果数据可以看出,ppAP 有效地提升了 AP 算法的性能,尤其是对大数据量非均衡数据集,ppAP 体现出了较大的优势.

Table 2 Comparison of clustering effectiveness

表 2 聚类效果比较

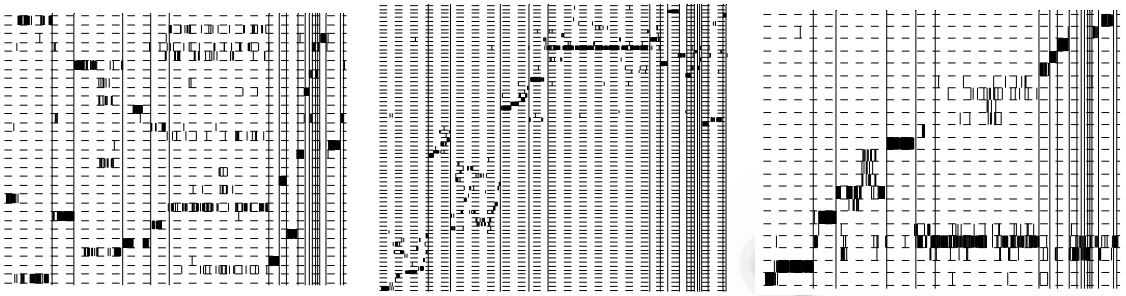
Dataset	Spectral	TribeMCL	CLUSS	BlastClust	AP	adAP	ppAP
SCOP_200	0.631 8	0.242 0	0.424 3	0.147 2	0.548 6	0.525 2	0.615 7
SCOP_512	0.443 3	0.194 3	0.377 8	0.129 3	0.328 5	0.315 5	0.514 8
COG_423	0.663 2	0.589 9	0.873 9	0.210 5	0.504 5	0.607 2	0.714 0
COG_487	0.670 7	0.512 5	0.782 9	0.189 7	0.566 0	0.612 7	0.721 5
COG_584	0.622 0	0.398 5	0.607 8	0.127 7	0.375 3	0.532 0	0.684 2
COG_765	0.632 8	0.329 5	0.565 6	0.133 9	0.380 8	0.630 5	0.750 1

为了更加清楚、明了地展示蛋白质序列的聚类结果,我们采用类似文献[4]提出的图形化方式展示聚类结果.图 3 和图 4 用图形化的方式展示了 Spectral,CLUSS,ppAP 这 3 种算法对 COG_584 和 COG_765 这两个数据集的聚类结果.图中两条横的虚线之间表示算法获得的一个分类,竖的长实线用来分隔数据集中“真实”的类,两条虚线之间出现的短竖线表示一条蛋白质序列.最终一个蛋白质序列出现在哪个位置要由它本身属于哪个家族和算法最终将它划分到哪个类来共同决定.从图 3 和图 4 中可以看出,COG_584 和 COG_765 这两个数据集本身类分布是比较“恶劣”的,并不均匀,既有很大的家族又有很多非常小的家族.

从图 3 中可以清晰地看出,ppAP 获得的聚类个数最接近于正确的分类个数,且聚类的结果较为紧密,尤其是对几个大的家族和一些较小的家族聚类结果基本接近正确的分类;CLUSS 算法获得的聚类个数较多且形成了很多成员个数很少的类,因此整体效果变得不佳,但其对其中最大的一个家族及其后面的几个小的家族聚类效果较好;谱聚类算法获得的聚类数也较为接近正确的分类个数,但其对几个大的家族的划分效果并不理想.

从图 4 中可以看出,谱聚类得出的聚类个数最接近正确的分类个数;但是从图中可以明显地看出,ppAP 得到的聚类结果要优于谱聚类.ppAP 对几个大类的数据除了少量的数据之外,大体上都划分到了一起;而谱聚类算

法将几个大类的数据基本上平分成了几个类,因此导致聚类结果不够紧密.从图 4 中可以看出,ppAP 的聚类结果形成了一个清晰的阶梯型.对 COG_765 数据集,CLUSS 算法同样产生了非常多的小类,并且包含一些“孤类”,因此整体效果不佳;但是它对第 3 个和第 4 个家族及后面的一些小的家族划分效果较好.

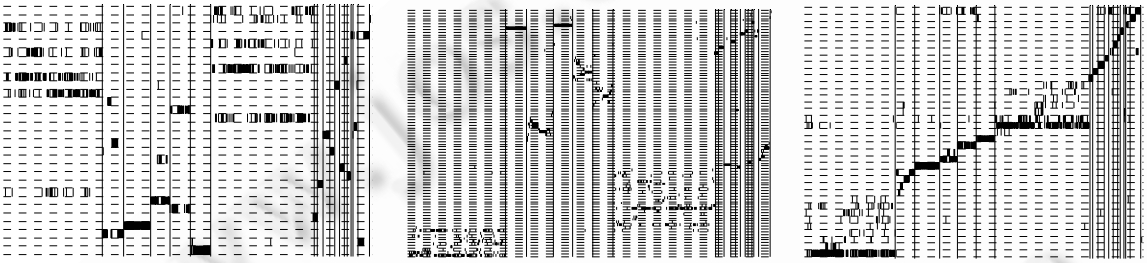


(a) Spectral clustering (b) CLUSS (c) ppAP

(a) 谱聚类 (b) CLUSS (c) ppAP

Fig.3 Clustering results on the COG_584 dataset

图 3 COG_584 数据集上的聚类结果



(a) Spectral clustering (b) CLUSS (c) ppAP

(a) 谱聚类 (b) CLUSS (c) ppAP

Fig.4 Clustering results on the COG_765 dataset

图 4 COG_765 数据集上的聚类结果

通过上述分析比较可以看出,ppAP 能够对蛋白质序列进行较好的聚类分析.对 6 个实验数据集的聚类结果表明,ppAP 在性能上优于或接近其他知名的蛋白质序列聚类算法.

5 结束语

随着基因组测序技术的发展,大量的未标注功能的蛋白质序列产生了.聚类方法通过分组不同功能的蛋白质序列,可以帮助预测未知功能的蛋白质序列,因此,目前蛋白质序列聚类研究是生物信息学研究中的一个重要领域.本文主要对蛋白质序列数据聚类方法进行研究,在本文中,首先详细地描述了最近出现的 AP 算法,并指出了在实际使用 AP 算法中出现的一些问题.对蛋白质序列进行聚类分析的一个重要的步骤就是计算蛋白质序列之间的相似度,为此,本文分析了现有的基于序列比对的相似度计算方法,并指出了这些方法的不足之处.然后,本文给出了一种改进的无比对相似度计算方法.为了解决这些问题,本文提出了一种有效的基于仿射传播聚类算法和后处理方法的蛋白质序列聚类方法(ppAP).与其他知名的蛋白质序列聚类算法进行对比,实验结果表明,本文给出的蛋白质序列聚类算法能够很好地对蛋白质序列进行聚类分析,在多个数据集上算法的聚类性能都优于或接近其他聚类算法.对蛋白质序列进行聚类分析是一个非常复杂的问题,并且由于目前序列数据总量非常庞大,因此对聚类算法提出了较高的要求.如何在给出更加高精度聚类算法的同时平衡运行时间和空间的耗

费,将是未来研究的重点.

致谢 在此,我们谨对电子科技大学生命科学学院张江博士表示诚挚的感谢.

References:

[1] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Journal of Software, 2008,19(1):48−61 (in Chinese with English abstract). http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20080106&flag=1 [doi: 10.3724/SP.J.1001.2008.0048]

[2] Qian WN, Zhou AY. Analyzing popular clustering algorithms from different viewpoints. Journal of Software, 2002,13(8): 1382−1394 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/1382.htm

[3] Zhao Y, Karypis G. Data clustering in life sciences. Molecular Biotechnology, 2005,31(1):55−80. [doi: 10.1385/MB:31:1:055]

[4] Paccanaro A, Casbon JA, Saqi MAS. Spectral clustering of protein sequences. Nucleic Acids Research, 2006,34(5):1571−1580. [doi: 10.1093/nar/gkj515]

[5] Enright AJ, Ouzounis CA. GeneRAGE: A robust algorithm for sequence clustering and domain detection. Bioinformatics, 2000, 16(5):451−457. [doi: 10.1093/bioinformatics/16.5.451]

[6] Wittkop T, Baumbach J, Lobo FP, Rahmann S. Large scale clustering of protein sequences with FORCE—A layout based heuristic for weighted cluster editing. BMC Bioinformatics, 2007,8:396. [doi: 10.1186/1471-2105-8-396]

[7] Yona G, Linial N, Linial M. ProtoMap: Automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. Proteins, 1999,37(3):360−378. [doi: 10.1002/(SICI)1097-0134(19991115)37:3<360::AID-PROT5>3.0.CO;2-Z]

[8] Krause A, Stoye J, Vingron M. The SYSTERS protein sequence cluster set. Nucleic Acids Research, 2000,28(1):270−272. [doi: 10.1093/nar/28.1.270]

[9] Pipenbacher P, Schliep A, Schneckener S, Schönhuth A, Schomburg D, Schrader R. ProClust: Improved clustering of protein sequences with an extended graph-based approach. Bioinformatics, 2002,18(Suppl. 2):S182−191. [doi: 10.1093/bioinformatics/18.suppl_2.S182]

[10] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology, 1990, 215(3):403−410.

[11] Enright AJ, Van Dongen S, Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research, 2002,30(7):1575−1584. [doi: 10.1093/nar/30.7.1575]

[12] Kelil A, Wang S, Brzezinski R, Fleury A. CLUSS: Clustering of protein sequences based on a new similarity measure. BMC Bioinformatics, 2007,8:286. [doi: 10.1186/1471-2105-8-286]

[13] Vinga S, Almeida J. Alignment-Free sequence comparison—A review. Bioinformatics, 2003,19(4):513−523. [doi: 10.1093/bioinformatics/btg005]

[14] Frey BJ, Dueck D. Clustering by passing messages between data points. Science, 2007,315(5814):972−976. [doi: 10.1126/science.1136800]

[15] Lai DR, Lu HT. Identification of community structure in complex networks using affinity propagation clustering method. Modern Physics Letter B, 2008,22(16):1547−1566. [doi: 10.1142/S0217984908016285]

[16] Du CH, Yang J, Wu Q, Li F. Integrating affinity propagation clustering method with linear discriminant analysis for face recognition. Optical Engineering, 2007,46(11):110501. [doi: 10.1117/1.280173]

[17] Xiao Y, Yu J. Semi-Supervised clustering based on affinity propagation. Journal of Software, 2008,19(11):2803−2813 (in Chinese with English abstract). http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20081103&flag=1 [doi: 10.3724/SP.J.1001.2008.2803]

[18] Wang KJ, Zhang JY, Li D, Zhang XN, Guo T. Adaptive affinity propagation clustering. Acta Automatica Sinica, 2007,33(12): 1242−1246 (in Chinese with English abstract).

[19] Varre JS, Delahaye JP, Rivals E. Transformation distances: A family of dissimilarity measures based on movements of segments. Bioinformatics, 1999,15(3):194−202. [doi: 10.1093/bioinformatics/15.3.194]

[20] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 1986,20:53−65.

[21] Wikipedia. http://en.wikipedia.org/wiki/Coefficient_of_variation

[22] Bolten E, Schliep A, Schneckener S, Schomburg D, Schrader R. Clustering protein sequences—Structure prediction by transitive homology. Bioinformatics, 2001,17(10):935−941. [doi: 10.1093/bioinformatics/17.10.935]

[23] Li YJ. A clustering algorithm based on maximal θ-distant subtrees. Pattern Recognition, 2007,40(5):1425−1431. [doi: 10.1016/j.patcog.2006.10.003]

[24] Tseng VS, Kao CP. Efficiently mining gene expression data via a novel parameterless clustering method. IEEE/ACM Trans. on Computational Biology and Bioinformatics, 2005,2(4):355−365. [doi: 10.1109/TCBB.2005.56]

[25] Conte LL, Ailey B, Hubbard TJP, Brenner SE, Murzin AG, Chothia C. SCOP: A structural classification of proteins database. Nucleic Acids Research, 2000,28(1):257−259.

[26] SCOP website. http://scop.mrc-lmb.cam.ac.uk/scop/

[27] COG. http://www.ncbi.nlm.nih.gov/COG/

[28] Ng A, Jordan M, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14. Boston: MIT Press, 2001. 849−856.

附中文参考文献:

[1] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48−61. http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20080106&flag=1 [doi: 10.3724/SP.J.1001.2008.0048]

[2] 钱卫宁,周傲英.从多角度分析现有聚类算法.软件学报,2002,13(8):1382−1394. http://www.jos.org.cn/1000-9825/13/1382.htm

[17] 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. http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20081103&flag=1 [doi: 10.3724/SP.J.1001.2008.2803]

[18] 王开军,张军英,李丹.自适应仿射传播聚类.自动化学报,2007,33(12):1242−1246.

唐东明(1979－),男,湖北宜昌人,博士,讲师,主要研究领域为模式识别,生物信息学.

杨凡(1979－),男,博士,CCF 学生会员,主要研究领域为生物信息学.

朱清新(1954－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为生物信息学,计算机图形与视觉,计算运筹学.

陈科(1979－),男,博士,主要研究领域为生物信息学.