文章编号：2096-1472(2017)-08-17-04

一种新的序列集保守信号识别算法

刘 畅
(国网冀北电力有限公司秦皇岛供电公司，河北 秦皇岛 066000)

摘 要：模体发现是计算机科学中的一个较为重要且具有一定挑战的问题，主要用于定位DNA序列集中的保守信号。首先，分析了已有的基于图聚类的模体发现算法MCL-WMR，讨论了它存在的两个缺陷。其次，针对这两个缺陷提出了MCL-WMR的改进算法iMCL-WMR。实验结果表明，所提DNA模体发现算法的时间性能好于所比较的算法MCL-WMR和qPMS9，能够在1个小时以内处理数百条输入序列，而且能够应对某些输入序列不含模体实例的测试数据。

关键词：保守信号；模体发现；图聚类
中图分类号：TP301.6 **文献标识码**：A

A New Algorithm for Identifying Conserved Signals in Sequence Sets

LIU Chang
(*Department of Information and Communication,Qinhuangdao Branch of State Grid Jibei Electric Power Company,Qinhuangdao* 066000,*China*)

Abstract: Motif discovery is an important and challenging issue in computer science,mainly used to locate conserved signals in a set of DNA sequences.At first,a graph clustering based motif discovery algorithm MCL-WMR is analyzed and its two drawbacks are discussed.Then,in order to overcome the two drawbacks,an improved algorithm of MCL-WMR,named as iMCL-WMR,is proposed.Experimental results show that,with a better time performance than the compared algorithms of MCL-WMR and qPMS9,the proposed algorithm can process hundreds of sequences within one hour and deal with any case that some input sequences do not contain motif instances.

Keywords:conserved signals;motif discovery;graph clustering

1 引言(Introduction)

序列集中保守的信号称为模体。模体发现就是在一组给定的序列集合中找出未知的模体，是很多重要的数据挖掘任务的第一步[1-3]。本文主要关注DNA序列集合中的模体发现问题[4]，可用于定位许多具有重要功能的基因区域。问题形式化描述如下[5]：

给定字符表{A,C,G,T}上的n条长为m的序列集合$S=\{s_1,s_2,\cdots,s_n\}$，目标是找出一个长为l(0<l<m)的字符串c，使得每条序列在s_i中都包含一个与c存在至多d(0≤d<l)个位置差异的长为l的字符串c_i。c称为一个(l,d)模体，c_i称为模体c的一个模体实例。后文称一个长为l的字符串为一个l-mer。

此问题是一个NP难问题[6]，当前已涌现了许多求解算法[7]。一类算法通过穷尽所有可能的l-mer来找出潜在的模体[8,9]。另一类算法将模体视为一个位置频率矩阵(Position Frequency Matrix,PFM)[10]，一般对初始的PFM进行迭代优化来得到模体。还有一类算法将模体发现问题抽象为图聚类问题，比如MCL-WMR[11]。

由于此问题的NP难特性，目前它仍然是一个开放性的难题；此外，当前待处理的模体发现的数据规模在变大[12]，许多现有算法在时间性能上达不到实用要求。本文通过分析和改进MCL-WMR，提出了一种新的模体发现算法，它的时间性能更为高效且能够处理某些输入序列不含模体实例的情况。

2 相关工作(Related works)

数据聚类是将一个给定的数据集中的数据项进行分组，使得同一组中的数据项高度相似并显著区分于其他组中的数据项。对于模体发现，由于同一转录因子的结合位点在某种程度上相互相似，采用合适的相似性度量的聚类算法可以将输入的DNA序列集合中的结合位点聚集在一起。然而，由于模体在序列中往往是高度退化的，定义合适的相似性度量和聚类模型对于在没有任何先验的情况下找到相关的结合位点是一个困难的任务。

模体的表示模型在设计模体发现算法中是至关重要的。一个最广为使用的模体表示模型是位置频率矩阵PFM。一个长度为l的模体的PFM是一个矩阵$M_{pfm}=[a_{bi}]_{4\times l}$，其中$b=\{A,C,G,T\}$，$i=1,\cdots,l$，并且对于任意的$i$有$\Sigma a_{bi}=1$。PFM表示一组对齐的等长位点中一个核苷酸在某个位置上出现的概率，它也表示了蛋白质与DNA进行交互的强度。

MCL是一个知名的图聚类算法[11]。它的基本原理是模拟

随机游走。考虑一个图中的聚类，聚类内部将含有许多边，而聚类之间将含有较少的边；这意味着，从图中的一个顶点出发随机游走到有边相连的另一个顶点时，更有可能停留在一个聚类内部，而跨越到另一个聚类的概率会比较低。通过在图中进行随机游走，可以发现流(flow)聚焦的地方，即聚类之所在。图中的随机游走使用马尔可夫链进行计算，即通过计算图的概率矩阵的连续的乘方来模拟流。MCL每次迭代中主要包含扩展和膨胀两个步骤，其中膨胀参数r决定了聚类的粒度。

MCL-WMR[11]是一个基于图聚类的模体发现算法，它的伪代码如下所示。它将输入的DNA序列集合转为一个带权图，然后采用MCL对图进行聚类从而得出模体。序列中每个l-mer被表示为一个顶点，在图的构建中保证与模体实例相对应的两个顶点间要有边相连接，这样模体发现转化为了在图中寻找大小为n的团。

Algorithm MCL-WMR

Input：l,d,S={$s1$,$s2$,…,s_n}

Output：a motif

1:build a graph G according to the given data set S

2:select a reference sequence sr from S randomly

3:for each l-mer x in sr do

4:generate a subgraph G' induced by x

5:cluster G' using MCL

6:for each cluster do

7:if the weight of the current cluster is above threshold then

8:verify that if the cluster contains a motif

9:if find a motif,output it

建图的具体过程为：(i)顶点集中的顶点$v_{i,j}$表示第i条序列位置j处的l-mer，对于每个i，j=1,2,…,m-l+1，所以共有n(m-l+1)个顶点；(ii)对于每对顶点$v_{i,j}$和$v_{i',j'}$($i\neq i'$)，当表示的两个l-mer的海明距离小于等于2d时，$v_{i,j}$和$v_{i',j'}$间有一条边相连；(iii)当有边相连的两个顶点的海明距离k满足$d<k\leqslant 2d$时，该边的权值设置为l-k，当k满足$k\leqslant d$时，该边的权值设置为10(l-k)。这种边的权值设置方法强化了海明距离较小的情况，也即海明距离越小意味着对应的两个子串越有可能是同一模体的两个实例。

在MCL-WMR算法中，为了减少MCL所处理的顶点数量，从输入序列集合S中任取一条参考序列s_r，然后由s_r中的每个l-mer x诱导一个子图G'，使得G'是由{S-s_r}中与x满足海明距离小于等于2d的所有l-mer组成。然后分别对G'用MCL进行聚类；由于任意两个模体实例满足海明距离小于等于2d，所以必然存在一个子图G'，其中包含由n个模体实例组成的团。

3 方法(Method)

3.1 算法思想

MCL-WMR存在如下的两个缺陷。

首先，MCL-WMR假设每条输入序列至少含有一个模体实例，因此不能应对有部分序列不含有模体实例的情况。在真实的情况中，生物学家产生的数据或经过初步处理的数据中并不能保证每条序列都含有模体的出现。此外，当模体以高度退化的形式出现在某些序列中时，它并不能显著地区分于背景中的子串。根据模体在序列中出现的情况，可以分为OOPS、ZOOPS和TCM三种模型，分别表示模体在每条序列中出现一次、模体在每条序列中出现零次或一次、模体在每条序列中出现零次或多次。显然，MCL-WMR只能应对OOPS模型的情况，而不能应对更符合实际情况的ZOOPS和TCM模型的情况。

其次，MCL-WMR不能应对规模较大的输入数据。MCL-WMR在处理经典的20条长为600的序列集时，有较好的时间性能。近年来，随着生物实验和测序技术的发展，染色质免疫共深沉与高通量测序相结合的ChIP-seq技术[12]可以使人们在基因组水平上进行模体发现，但处理的数据量包含数百条甚至更多条序列。当数据规模变大时，即使MCL-WMR对多个子图分别进行处理，但每个子图要处理的数据量依然很大，使得MCL聚类的时间效率变得很低。

为了解决这两个问题，本文对MCL-WMR进行改进，提出一个称为iMCL-WMR的算法。基本思想如下：评估输入序列中每个l-mer的信号强度，将信号较强的l-mer视为潜在的模体实例；由每个潜在的模体实例诱导一个子图G'，使得G'中有边相连的两个l-mer间的海明距离小于等于d；对G'进行聚类，对得到的各个聚类进行求精，最后选择得分最大的进行输出。

iMCL-WMR不受限于每条序列必须包含一个模体实例的OOPS模型，它能够适应ZOOPS和TCM模型的情况。此外，当处理的数据规模较大时，存在一些较为相似的模体实例，即它们之间的海明距离小于等于d；因此，在由一个潜在的模体实例诱导一个子图时，iMCL-WMR限定有边相连的两个l-mer间的海明距离小于等于d，这样iMCL-WMR中的子图的数据规模将小于MCL-WMR中的子图的数据规模，从而提高MCL聚类的时间效率。

3.2 划分子序列集合

首先，评估各个l-mer的信号强度。令$s_{i,j}$表示第i条序列s_i位置j处的l-mer。一个任意的l-mer x与给定的序列s_i的距离$dis(x,s_i)$，定义为x与si中的l-mer的可能的最小海明距离，如式(1)所示；其中，$d_H(x,s_{i,j})$表示两个l-mer的海明距离，即两个l-mer对齐后字符不相等的位置的个数。

$$dis(x,s_i)=\min_{j=1\ldots m-l+1} d_H(x,s_{i,j}) \tag{1}$$

令$SD(x)$表示输入序列中的一个l-mer x的信号强度，定义为x与各个输入序列s_i中最为相似的l-mer的总的相同的位置个数，如式(2)所示。

$$SD(x)=\sum_{i=1}^{n} l-dis\left(x,s_i\right) \tag{2}$$

一个l-mer x的信号强度越大，它越有可能是一个潜在的

模体实例。这可以从模体发现的问题属性来解释：模体的各个实例是由同一模体变异而产生的，所以各个实例间是相互相似的；这样，如果一个l−mer x是模体实例，那么部分序列中的模体实例将与其非常相似，使得x的信号强度显著地大于背景中的或随机的l−mer的信号强度。

计算出输入序列中所有的l−mer $s_{i,j}$的信号强度$SD(s_{i,j})$后，取其均值和标准差，分别记为μ和σ。令L表示输入序列中所有的l−mer $s_{i,j}$的集合，即$L=\{s_{i,j}:1\leqslant i\leqslant n,1\leqslant j\leqslant m\}$。将$L$中的$l$−mer近似成正态分布，如图1所示，横轴表示信号强度，纵轴表示对应信号强度的l−mer的数量。基于此，将L划分成L_1、L_2和L_3三部分。

$$L_1=\left\{s_{ij}:1\leq i\leq n,1\leq j\leq m,SD(s_{ij})<\mu-\sigma\right\} \tag{3}$$

$$L_2=\left\{s_{ij}:1\leq i\leq n,1\leq j\leq m,\mu-\sigma\leq SD(s_{ij})\leq\mu+\sigma\right\} \tag{4}$$

$$L_3=\left\{s_{ij}:1\leq i\leq n,1\leq j\leq m,SD(s_{ij})>\mu+\sigma\right\} \tag{5}$$

根据序列中l−mer的信号强度的定义，对所有子序列的集合的划分解释如下。L_1是背景序列区，即它对应很小的信号强度，包含模体实例的概率很低。L_3是潜在的模体实例区，即它对应很大的信号强度，包含模体实例的概率很高。L_2是模体实例和背景序列的混合区，它既包含模体实例也包含背景序列，且难以进行区分。基于这个划分，iMCL−WMR在构建图时：一方面，仅使用L_2和L_3中的元素构建图G，这样可以减小数据规模；另一方面，仅使用L_3中的元素诱导子图G'，这样可以控制子图的数量。

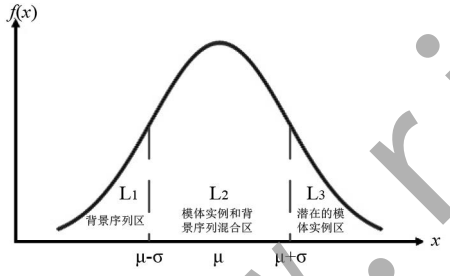

图1 子序列划分示意图

Fig.1 Illustration of sub−sequence partition

3.3 聚类求精

为了保证良好的时间性能，iMCL−WMR执行聚类的子图中并不一定包含全部的模体实例，这样得到的聚类C是一个包含部分模体实例的集合。因此，需要对C进行求精，得到全部的模体实例及模体。求精算法如下。

Algorithm RefineCluster

Input：$l,d,S=\{s_1,s_2,\cdots,s_n\},C$

Output：a motif

1:generate a consensus c according to aligned elements in C

2:score←∞,curScore←0

3:while curScore>score do

4:score←curScore,C←Φ

5:for i←1 to n do

6.put an l−mer x in s_i with $dH(c,x)=dis(c,s_i)$ to C

7:generate a new consensus c according to aligned elements in C

8:curScore ← IC(c)

9:output c and its score curScore

求精算法迭代地对当前的模体进行更新，直到得到信息量最高的模体。模体c的信息量记为$IC(c)$，计算方法如下所示，其中f_{ij}表示C中l−mer对齐后第j列的字符i的出现频率，b_i表示字符i的背景频率。

$$IC(c)=\sum_{j=1}^{l}\sum_{i\in\{A,C,G,T\}}f_{ij}\ln\frac{f_{ij}}{b_i} \tag{6}$$

3.4 整体算法

iMCL−WMR算法的伪代码描述如下。

Algorithm iMCL−WMR

Input:$l,d,S=\{s_1,s_2,\cdots,s_n\}$

Output:a motif

1:compute the signal degree for each l−mer in S

2:divide all l−mers in S to three sets L_1,L_2 and L_3

3:build a graph G using the l−mers in L_2 and L_3

4:for each l−mer x in L_3 do

5:generate a subgraph G' induced by x

6:cluster G' using MCL

7:for each obtained cluster C do

8:refine C using algorithm RefineCluster

9:obtain a motif c and its score curScore

10:rank the obtained motifs according to their scores

11:output the motif with maximum score

算法第1行伪代码计算输入序列中各个l−mer的信号强度。依据信号强度，第2行伪代码将输入序列中的l−mer划分成L_1、L_2和L_3三个集合。第3行伪代码使用L_2和L_3中的l−mer构建一个图G，注意iMCL−WMR要求两个l−mer间的海明距离k小于等于d时对应的两个顶点才有边相连接；边的权值设置为$(d-k)(l-k)$，这比MCL−WMR的权值设置方法更为精细，使得海明距离越小时越以更大的概率视为一对模体实例。第4—9行伪代码对L_3中的各个l−mer诱导的子图依次进行聚类求精，得到多个模体。第10—11行伪代码对得到的多个模体依照其得分进行排序，最终输出得分最高的模体。

4 实验结果(Experimental results)

首先，使用经典的模拟数据集(随机生成20条长为600的DNA序列，然后在每条序列中植入一个(l,d)模体，使得模体与模体实例间最大差异d个位置)，比较所提算法iMCL−WMR与现有算法的运行时间和识别准确率。除了与MCL−WMR进行比较之外，又选取了qPMS9作为比较算法，它是最近提出的算法并且是时间性能最好的精确算法。所有算法都是用C++实现的，并且运行于相同的平台(3.0GHz的处理器和4GB的内存)。选用的(l,d)为挑战问题实例，即(9,2)、(11,3)、(13,4)、(15,5)、(17,6)、(19,7)、(21,8)、(23,9)、(25,10)；对于每个(l,d)问题实例，随机生成了六组数据，实验结果取

均值。识别准确率用核苷酸水平的性能系数来衡量，它等于预测出的模体中的核苷酸与真实的模体中的核苷酸的交集的大小除以其交并集的大小。

表1 不同(l,d)问题实例上的结果

Tab.1 Results on different(l,d) problem instances

(l,d)	iMCL-WMR	MCL-WMR	qPMS9
(9,2)	1.00 (20s)	0.93 (2.3m)	1.00 (1s)
(11,3)	1.00 (42s)	1.00 (5.9m)	1.00 (2s)
(13,4)	0.92 (1.2m)	1.00 (9.8m)	0.97 (6s)
(15,5)	1.00 (4.3m)	0.98 (14.4m)	1.00 (32s)
(17,6)	1.00 (6.8m)	1.00 (18.3m)	1.00 (2.4m)
(19,7)	0.95 (11.9m)	1.00 (28.6m)	1.00 (12.7m)
(21,8)	1.00 (15.4m)	0.96 (45.9m)	0.98 (43.6m)
(23,9)	0.96 (19.3m)	1.00 (65.4m)	1.00 (2.2h)
(25,10)	0.98 (33.0m)	0.97 (78.8m)	1.00 (6.1h)

表1给出了这些(l,d)问题实例上的实验结果，其中s表示秒、m表示分钟、h表示小时。整体来看，三个算法的识别准确率是相当的；qPMS9的识别准确率略好于iMCL-WMR和MCL-WMR的识别准确率，因为它是一个精确算法，穷尽地遍历了所有的候选模体。对于时间性能，iMCL-WMR在整体上好于MCL-WMR和qPMS9；qPMS9在求解小的(l,d)问题实例时具有优势，这是因为求解小的(l,d)问题实例需要穷尽遍历的候选模体的数量较少，但在求解大的(l,d)问题实例时它的时间性能很差；iMCL-WMR的时间性能好于MCL-WMR的原因是对MCL聚类的子图进行了降维处理。

其次，固定(l,d)为(15,5)问题实例以及每条序列的长度为600，变化序列数量n，同时只在80%的序列中植入模体实例。由于MCL-WMR不支持非OOPS模型的情况，这里只与qPMS9进行了比较。实验结果如表2所示。可以发现，在求解非OOPS模型的输入数据时，所提算法iMCL-WMR的时间性能显著地好于qPMS9的时间性能。这是因为，iMCL-WMR依据信号强度划分输入序列中的l-mer，再诱导子图进行聚类的方法可以很自然地处理非OOPS模型的情况；qPMS9是将输入数据转化成多个OOPS模型的数据再进行穷举，所以当数据规模较大时，时间性能很差。

表2 不同规模数据集上的结果

Tab.2 Results on data of different scale

n	iMCL-WMR	qPMS9
50	5.6m	2.4h
100	11.2m	18.9h
200	18.6m	>24h
300	25.4m	>24h
400	32.7m	>24h
500	43.2m	>24h

5 结论(Conclusion)

本文提出了一种新的模体发现算法iMCL-WMR。实验结果表明，一方面，iMCL-WMR的时间性能好于所比较的算法MCL-WMR和qPMS9，能够在1个小时以内处理数百条输入序列；另一方面，iMCL-WMR能够应对某些输入序列不含模体实例的测试数据。总之，iMCL-WMR在处理较大规模和更符合真实场景的DNA序列数据集时具有较好的实用性。

参考文献(References)

[1] Wong KC,et al.A comparison study for DNA motif modeling on protein binding microarray[J].IEEE/ACM Transactions on Computational Biology and Bioinformatics,2016,13(2):261-271.

[2] Kranjc J,et al.ClowdFlows:Online workflows for distributed big data mining[J].Future Generation Computer Systems,2017,68:38-58.

[3] Liu B,et al.Efficient motif discovery for large-scale time series in healthcare[J].IEEE Transactions on Industrial Informatics.2015,11(3):583-590.

[4] Rajagopal N,et al.High-throughput mapping of regulatory DNA[J].Nature biotechnology,2016,34(2):167.

[5] Pevzner PA,Sze SH.Combinatorial approaches to finding subtle signals in DNA sequences[C].In:Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology,California,2000:269-278.

[6] Evans PA,Smith AD,Wareham HT.On the complexity of finding common approximate substrings[J].Theory Computer Science,2003,306:407-430.

[7] Zambelli F,Pesole G,Pavesi G.Motif discovery and transcription factor binding sites before and after the next generation sequencing era[J].Briefings in Bioinformatics,2013,14(2):225-237.

[8] Al-Okaily A,Huang CH.ET-motif:solving the exact(l,d)-planted motif problem using error tree structure[J].Journal of Computational Biology,2016,23(7):615-623.

[9] Nicolae M,Rajasekaran S.qPMS9:an efficient algorithm for querum planted motif search[J].Scientific Reports,2015,5:7813.

[10] Machanick P,Bailey TL.MEME-ChIP: motif analysis of large DNA datasets[J]. Bioinformatics,2011,27(12):1696-1697.

[11] Boucher C,Brown D,Church P.A graph clustering approach to weak motif recognition[C].In:Proceedings of the 7th International Workshop on Algorithms in Bioinformatics,Philadelphia,PA,USA,2007:149-160.

[12] 高山,等.下一代测序中ChIP-seq数据的处理与分析[J].遗传,2012,34(6):773-783.

作者简介：

刘 畅(1974-)，男，本科，工程师.研究领域：软件开发.