Computer Science and Application 计算机科学与应用, 2012, 2, 17-21
http://dx.doi.org/10.12677/csa.2012.21004 Published Online March 2012 (http://www.hanspub.org/journal/csa)

An Improved KNN Algorithm for Haplotype Reconstruction Problem

Tao Zhuang, Hong Liu

School of Computer Science and Technology, Shandong University, Jinan
Email: zhtao663@163.com

Received: Dec. 1st, 2011; revised: Dec. 29th, 2011; accepted: Jan. 14th, 2012

Abstract: Single Nucleotide Polymorphisms (SNPs) is a single base pair position in genomic DNA where different nucleotide variants exist in some population, and is considered as the most frequent form of human genetic variants. The haplotype is composed of the SNP which was found to contain more genetic information. Study it plays an important role in the diagnosis of disease and drugs design. Haplotype reconstruction is to reconstruct a pair of haplotypes from localized polymorphism data got through short genome fragment assembly. In this paper, a new clustering method, based on KNN and PSO algorithms, is proposed to solve haplotype reconstruction problem. In the end, it will be used simulation data and real biological data to test the proposed algorithm, and the results show that the proposed method is feasible.

Keywords: SNP; Haplotype Assembly; Cluster; KNN; PSO

基于改进的 *k* 最近邻算法的单体型重建问题

庄　涛，刘　宏

山东大学计算机科学与技术学院，济南
Email: zhtao663@163.com

收稿日期：2011 年 12 月 1 日；修回日期：2011 年 12 月 29 日；录用日期：2012 年 1 月 14 日

摘　要：单核苷酸多态性(SNPs)是人类遗传变异中最显著的一种形式，是一个物种中 DNA 序列中某个位点上的碱基变化。人们发现由单核苷酸组成的单体型比单一的单核苷酸包含更多的生物遗传信息，因此研究单体型对于诊断疾病和药物研制有着重要作用。单体型重建就是对由 SNP 片段组成的基因片段进行组装，从而构造出原来的一对单体型。本文在 *k* 最近邻和粒子群算法的基础上，提出一种解决单体型重建问题的一种聚类算法。最后，本文将用模拟数据和真实数据来检验本文所提出的算法，结果证明所提出的算法可行。

关键词：单核苷酸多态性；单体型重建；聚类；*k* 最近邻算法；粒子群算法

1. 引言

随着人类基因组图谱的基本完成[1]，人们对遗传的差异性、由基因突变引起的疾病复杂性有了更精确的阐释[2]。现在人们普遍认为，DNA 序列中少数的差异是导致遗传疾病的主要原因。单核苷酸多态性(SNPs)，DNA 某一位置碱基的变化[3]，被认为是一个物种不同个体表型的主要遗传来源[4]。研究 SNPs 对基因的研究、遗传疾病的诊断和药物研制有着重要作用。人类的 DNA 序列是按染色体成对出现的，每一条染色体上 SNP 位点上的碱基序列叫做单体型，所以人类等二倍体生物都有一对单体型。在医学研究中，单体型数据通常比单个 SNP 携带更多的信息。基于单体型在遗传分析上的重要性，现在人们较为关注的是单体型的检测问题。

令人遗憾的是在当前的实验技术下，直接通过生物实验手段来得到单体型既费时又费力，因此利用计算机技术，通过计算的方式来获得单体型具有很好的实际应用价值。目前检测单体型的方法主要有两种：个体单体型检测，也称为单体型重建；群体单体型检测，也称为单体型推断。本文主要对前者进行研究。SNP 位点是一个物种的基因组 DNA 序列中不同个体可能出现不同碱基的位置。位于一个 SNP 位点的碱基称为等位基因。对于任意一个 SNP 位点来说，若两条同源染色体上的碱基相同，则称为纯和位点；若不相同，则称为杂合位点。几乎所有的 SNP 位点上的碱基都只有两种取值，为方便起见，我们用字符集{0,1,-}上的字符序列来表示单体型，而不必用真正的碱基字符，其中“-”表示该位点的取值未知，被称为空。因此单体型可看作是一个字符串序列。

尽管科学技术有了很大的进步，但是由于技术的限制、基因组的测序错误以及测试样本的污染等原因，通过实验手段直接得到单体型是很困难的且代价极为昂贵的。因此人们常利用计算的方式，设计一个准确有效地算法来解决单体型重建问题。

依赖于不同的数据错误类型，主要有几种不同的解决模型。其中主要有最少片段删除模型(MFR)、最少 SNP 去除模型(MSR)和另外一种被普遍应用的模型——最少错误纠正模型(MER)[4,5]。其中 MER 模型首先被 Lippert 等人证明是 NP-hard 问题[5]。其它模型，例如 LHR，MEC/GI，WMLF，WMEC 等也被用到[6,7]。实际上单体型重建问题可以被看作是一个聚类问题。一种启发式的聚类算法在文章[8,9]中已被应用。本文中，我们将给出另外一种基于 k 最近邻算法和粒子群优化算法的具有较好结果和效率的启发式的聚类方法。

本文其他部分的组织结构如下：在第二部分中，我们将给出单体型重建问题的形式化描述，给出一些必要的字符、公式定义。我们的方法将在第三部分给出。实验结果将在第四部分给出。最后，我们将总结本文。

2. 符号、公式定义

假设给定来自某对同源染色体的 m 条 SNP 片段，每条 SNP 片段对应的单体型长度为 n。为了描述方便，在下文，所提到的片段也指 SNP 片段。我们定义一个 $m \times n$ 的 SNP 矩阵 $\boldsymbol{M} = [m_{ij}]$，$0 < i \le m$，$0 < j \le n$，每个元素的值为 0、1 或-矩阵 $\boldsymbol{M}$ 中的每一行代表一条 SNP 片段，每一列代表一个 SNP 位点。

给出变量 x，$y \in \{0,1,-\}$，并给出如下定义：

$$s(x,y) = \begin{cases} 1 & \text{如果 } x \ne -,\ y \ne - \text{ 且 } x = y \\ 0 & \text{其他} \end{cases} \tag{1}$$

$$d(x,y) = \begin{cases} 1 & \text{如果} x \ne -,\ y \ne - \text{ 且 } x \ne y \\ 0 & \text{其他} \end{cases} \tag{2}$$

对于两条 SNP 片段 $m_i = m_{i1},\cdots,m_{in}$ 和 $m_j = m_{j1},\cdots,m_{jn}$，给出如下定义：

$$S(m_i,m_j) = \sum_{k=1}^{n} s(m_{ik},m_{jk}) \tag{3}$$

$$D(m_i,m_j) = \sum_{k=1}^{n} d(m_{ik},m_{jk}) \tag{4}$$

如果 $D(m_i,m_j) > 0$，称这两条片段冲突，否则称之为兼容。片段和单体型之间的距离也类似给出定义。如果所有片段没有数据错误，则矩阵 $\boldsymbol{M}$ 中的行可以被分为两个不相交的子集，每个子集中的所有行相容且决定一条单体型。这时，称矩阵 $\boldsymbol{M}$ 是可行的，否则是不可行的。

定义 $P = (C_1,C_2)$，集合 C_1 和集合 C_2 代表片段划分到的两个集合。定义 $h_1 = (h_{11},h_{12},\cdots,h_{1n})$ 和 $h_2 = (h_{21},h_{22},\cdots,h_{2n})$ 为一对原始的单体型，用 $h' = (h'_1,h'_2)$ 代表通过算法构造的单体型。用算法对片段进行分类后，可以通过叠加同一集合中的片段构造 h'_1。

定义 $N_j^0(C)$ 表示集合 C 中所有片段第 j 列中 0 的个数。相同的，$N_j^1(C)$ 代表 1 的个数。因此，单体型根据以下方法进行构造：

$$h_{ij} = \begin{cases} 0 & \text{如果 } N_j^0(C_i) \ge N_j^1(C_i) \\ & \quad \text{且 } N_j^0(C_i) \ne 0 \\ 1 & \text{如果 } N_j^1(C_i) \ge N_j^0(C_i) \\ - & \text{如果 } N_j^0(C_i) = N_j^1(C_i) = 0 \end{cases} \tag{5}$$

其中 $i \in \{1,2\}$，$j = 1, 2,\cdots,n$。

最后，用重建率(*RR*)来衡量单体型重建的正确度。重建率说明了重新构建的单体型 $h' = (h'_1,h'_2)$ 与原始单体型 $h = (h_1,h_2)$ 之间的相似程度。定义如下：

$$RR(h,h') = 1 - \frac{\min(r_{11}+r_{22},r_{12}+r_{21})}{2n} \tag{6}$$

其中 $r_{ij} = D(h_i,h'_j)$，$i = 1,2$；$j = 1,2$ 并且

$$D\left(h_i,h'_j\right)=\sum_{k=1}^{n}d\left(h_{ik},h'_{jk}\right)$$

3. 算法

本节将给出解决单体型重建问题的启发式的聚类算法。首先，介绍 k 最近邻算法和粒子群算法。

粒子群算法也称为粒子群优化算法(Particle Swarm Optimization)，缩写为 PSO，是近年来发展起来的一种新的进化算法。PSO 算法和遗传算法相似，也是从随机解出发，通过迭代寻找最优解；也是通过适应度来评价解的品质，但比遗传算法规则更为简单，本算法没有遗传算法的“交叉”和“变异”操作，本算法通过追随当前搜索到的最优值来寻找全局最优。与遗传算法相比较，粒子群算法在大多数情况下，能较快的得到最优化的解[10]。

k 最近邻(k-Nearest Neighbor，KNN)分类算法，是较成熟和较简单的聚类分析算法之一。该方法的思路是：选取一个待分类样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本，如果这 k 个中的大多数样本属于某一个类别，则该待分类样本也属于这个类别。一种启发式的聚类算法已经在文章[8]中被提到，其数学形式表示如下：

$$\max\left\{\left(S\left(f_p,C_1\right)+D\left(f_p,C_2\right)\right),\left(S\left(f_p,C_2+D\left(f_p,C_1\right)\right)\right)\right\}\\ \text{and }\forall f_p\in C_1\text{ and }\forall f_p\in C_2 \tag{7}$$

通过这个公式，可以在每次迭代的过程中，将所有的片段分到相应的集合中。但是在某些情况下，可以通过理论及实验证明，(7)式并不准确。例如，C_1 中片段的数量不等于 C_2 中片段的数量，(7)式就显得没有多少意义了。为此，本文提出了一种改进算法。实验结果表明，该改进算法能取得更好更精确的结果，同时也提高了执行效率。

PHKN 算法

将粒子群算法和部分的 k 最近邻算法结合起来达到优化结果的目的，本文称之为 PHKN(英文)算法。首先，计算出任意两条 SNP 片段之间的距离，找出距离最大者，并将相对应的两条片段分别放到集合 C_1 和 C_2 中，作为集合划分的初始值。接下来，在每次分配过程中，通过粒子群算法为未划分的每一条片段分别从集合 C_1 和 C_2 中选取 k'条片段，这是本算法的关键所在。其中 k'的取值有以下公式决定：

$$k'=\left\lfloor \log_{1.5}\min\left\{n_{C_1},n_{C_2}\right\}\right\rfloor+1 \tag{8}$$

假设未划分的片段为 f_p，通过公式(3)和公式(4)分别计算出 f_p 与从 C_1、C_2 中选取的 k'条片段的距离 $d\left(f_p,C_1\right)$、$d\left(f_p,C_2\right)$。比较 $d\left(f_p,C_1\right)$ 与 $d\left(f_p,C_2\right)$ 的大小，若 $d\left(f_p,C_1\right)>d\left(f_p,C_2\right)$，将 f_p 划分到集合 C_1，否则我们将 f_p 划分到 C_2，更新集合 C_1、C_2，进入下一次迭代，直至所有片段被划分到两个集合为止。虽然选择了 k'条片段，但是主要原理还是与 k 最近邻算法有区别的，顶多是用了 k 最近邻算法的思想，加之在 k'条片段的选取中用到了粒子群算法，所以称本文的方法为 PHKN 算法。算法的详细步骤在表 1 中给出。

为了使得以上程序获得更加准确的结果，把 h'作为划分集合的初始值并且重复算法 PHKN 直到 h'不再变化或者算法被运行 100 次为止。

4. 实验结果

试验中采用真实数据和模拟数据来检验算法的准确性。并在中央处理器为 Intel Pentium IV 2.33 GHz，内存为 2 GB 的微机系统上用 Java 语言运行。

4.1. 模拟数据

首先，用模拟数据来测试 PHKN 算法。首先，随机生成 10 条不同的单体型，每条单体型长度为 60，并通过相似度参数 s 来生成对应的 10 条单体型，其中

Table 1. Pseudo code for the PHKN algorithm
表 1. PHKN 算法的伪代码

输入：$m\times n$ 的 SNP 矩阵 $\boldsymbol{M}$
输出：一对单体型 $h'=\left(h'_1,h'_2\right)$
步骤 1：寻找距离最大的两条片段 m_1 和 m_2，作为初始值放到集合 C_1 和 C_2 中。
步骤 2：初始化参数，种群大小 N，学习因子 C_1、C_2，粒子大小(SNP 片段的数量)k'以及最大迭代次数 GN。
步骤 3：运行粒子群算法，分别为每条片段从每个集合中找出最优的 k'条片段。
步骤 4：如果 $\left(S\left(f_p,C_1^{(k')}\right)+D\left(f_p,C_2^{(k')}\right)\right)\geq\left(S\left(f_p,C_2^{(k')}\right)+D\left(f_p,C_1^{(k')}\right)\right)$ 则 $C_1=C_1\cup f_i$； 否则 $C_2=C_2\cup f_i$。
步骤 5：如果仍有剩余的片段，则转到步骤 2，否则转到步骤 6。
步骤 6：停止，通过划分的两个集合产生一对单体型。

s 表示一对单体型中两条单体型之间的相似度[11]。然后采用著名的 shotgun 测序模拟数据生成器 Celsim 来生成实验所需片段。通过设置参数片段数 $m = 100$；$s = 0.5$；SNP 缺失率 g 分别为 0.25；0.5 和 0.75；错误率 e 分别为 0.1，0.2，0.3，0.4 来产生每对单体型的 12 个实例。然后用以上相同的参数设置，除了 s 设置为 0 以外，产生另外 120 个实例。PHKN 算法运行模拟数据的结果显示在图 1 中，纵坐标代表重建率(RR)，横坐标代表错误率(e)。

图 1 中的(a)、(b)图分别是针对相似度参数 $s = 0.5$ 和 $s = 0$ 时，该算法在不同的错误率，不同缺失率下相对应的结果。图 1 表明单体型之间的相似度越高，重建率越高。同时也表明随着片段错误率、SNP 缺失率的增大，算法的重建率逐渐降低。

4.2. 真实数据

实验中用到的真实数据与文献[6]一样，采用来自公开数据库的真实的单体型，该数据来自于国际人类基因组单体型图计划[12]2006 年 7 月发布的数据文件 genotypes_chrl_CEU_r21_nr_fwd_phased.gz，该文件中包含了 CEPH 样本(祖籍是北欧或西欧的美国犹他州人)中 60 个个体的单体型，每个单体型有 SNP 位点 193,333 个，本文实验随机选择一个个体指定长度的一对单体型。然后我们采用著名的 shotgun 测序模拟数据生成器 Celsim 来生成实验所需片段。其中所需设置的参数 m：40，160，300；g：0.25，0.5 和 0.75；e：0.1，0.2，0.3，0.4。表 2 是利用真实数据，在相同的条件下，把本文提出的算法与文献[8]中提到的算法进行比较的实验结果。从表 2 中可以看出，在相同的缺失率、错误率的情况下，本文提出的算法能得到更好的实验结果。尤其是在错误率很大的情况下，该算法较文献[8]依然能取得较好的实验结果。

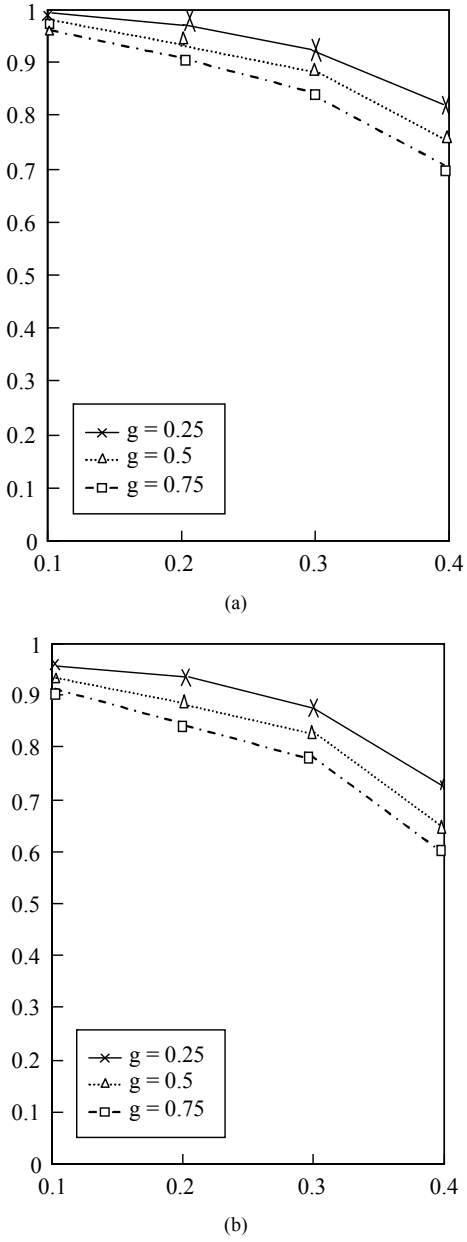

Figure 1. The reconstruction rate by the PHKN algorithm under different parameters: (a) $s = 0.5$; (b) $s = 0$

图 1. 不同参数下，PHKN 算法的重建率：(a) $s = 0.5$；(b) $s = 0$

Table 2. The comparative results of two algorithms

表 2. 两种算法的比较结果

g	m	$e = 0.1$		$e = 0.2$		$e = 0.3$		$e = 0.4$	
		Clustering	PHKN	Clustering	PHKN	Clustering	PHKN	Clustering	PHKN
0.25	40	0.967	0.980	0.946	0.958	0.901	0.919	0.752	0.764
	160	0.978	0.982	0.951	0.962	0.916	0.923	0.753	0.767
	300	0.980	0.990	0.959	0.963	0.931	0.945	0.764	0.786
0.5	40	0.932	0.949	0.917	0.935	0.892	0.905	0.688	0.710
	160	0.957	0.960	0.925	0.942	0.896	0.910	0.690	0.725
	300	0.960	0.970	0.951	0.958	0.903	0.916	0.696	0.733
0.75	40	0.920	0.932	0.901	0.911	0.802	0.819	0.637	0.652
	160	0.932	0.944	0.909	0.925	0.807	0.823	0.642	0.656
	300	0.940	0.954	0.911	0.930	0.812	0.833	0.645	0.668

5. 结论

本文设计了一种启发式的数据聚类算法，从两个集合中同时选择 k'条片段作为片段划分的依据是对文献[8]的改进，通过采用模拟数据和真实数据检验了改进算法的有效性。虽然经过改进，算法的准确度和执行效率有了很大提高，但仍然不能得到最优化的解，但是对单体型重建问题提供了一种快捷有效的解决方案。因此，在未来的研究中，将继续采用类似的方法来解决该问题。

参考文献 (References)

[1] J. C. Venter, M. D. Adams, et al. The sequence of the human genome. Science, 2001, 291(5507): 1304-1351.

[2] M. R. Hoehe, K. Kopke, B. Wendel, et al. Sequence variability and candidate gene analysis in complex disease: Association of μ opioid receptor gene variation with substance dependence. Human Molecular Genetics, 2000, 9(19): 2895-2908.

[3] The International HapMap Consortium. A haplotype map of the human genome. Nature, 2005, 437: 1299-1320.

[4] Z. Li, W. Zhou, X. Zhang and L. Chen. A parsimonious tree-grow method for haplotype inference. Bioinformatics, 2005, 21(17): 3475-3481.

[5] R. Lippert, R. Schwartz, G. Lancia and S. Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. Briefings in Bioinformatics, 2002, 3(1): 23-31.

[6] R. S. Wang, L. Y. Wu, Z. P. Li and X. S. Zhang. Haplotype reconstruction from SNP fragments by minimum error correction. Bioinformatics, 2005, 21(10): 2456-2462.

[7] X. S. Zhang, R. S. Wang. Models and algorithms for haplotyping problem. Current Bioinformatic, 2006, 1(1): 105-114.

[8] C. Eslahchi, M. Sadeghi, H. Pezeshk, M. Kargar and H. Poormohammadi. Haplotyping problem, a clustering approach, numerical analysis and applied mathematics. International Conference, 2007, 936: 185-190.

[9] Y. Wang, E. Feng and R. S. Wang. A clustering algorithm based on two distance functions for MEC model. Computational Biology and Chemistry, 2007, 31(2): 148-150.

[10] J. Kennedy, R. C. Eberhart. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, 27 November 1995-1 December 1995, 1942-1948.

[11] W. Y. Qian, Y. J. Yang, N. N. Yang and C. Li. Particle swarm optimization for SNP haplotype reconstruction problem. Applied Mathematics and Computation, 2008, 196(1): 266-272.

[12] The International HapMap Consortium. The international HapMap project. Nature, 2003, 426(6968): 789-796.

