基于彩色编码技术的准种重建算法

黄 丹[1] 吴璟莉[1,2,3]

(广西师范大学计算机科学与信息工程学院 广西 桂林 541004)[1]

(广西师范大学广西多源信息挖掘与安全重点实验室 广西 桂林 541004)[2]

(广西区域多源信息集成与智能处理协同创新中心 广西 桂林 541004)[3]

摘 要 求解病毒准种单体型有助于了解其基因结构特点,对疫苗的研制及抗病毒治疗具有重要意义。文中通过引入模糊距离,构造一种带权的片段冲突图,并提出了基于彩色编码技术的病毒准种单体型重建算法CWSS。CWSS算法先根据给定阈值对片段冲突图进行预处理;然后根据顶点的边权和及饱和度取值为图中顶点着色,着色遵循相邻顶点颜色相异的原则,直至所有顶点完成着色;最后将相同颜色的顶点片段进行组装,得到准种单体型。CWSS算法的时间复杂度为$O(m^2n+mn)$。采用模拟测序片段数据进行实验测试,对CWSS算法和Dsatur算法的重建性能和质量进行对比分析。实验结果显示,相比于Dsatur算法,CWSS算法能获得更准确的准种单体型,具有更高的重建性能。

关键词 准种,单体型,带权图,彩色编码,模糊距离

中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.02.046

Quasispecies Reconstruction Algorithm Based on Color Coding Technology

HUANG Dan[1] WU Jing-li[1,2,3]

(College of Computer Science and Information Technology, Guangxi Normal University, Guilin, Guangxi 541004, China)[1]

(Guangxi Key Laboratory of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi 541004, China)[2]

(Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, Guilin, Guangxi 541004, China)[3]

Abstract The reconstruction of viral quasispecies haplotypes contributes to know about the structure of viral gene-tic, and is of great significance for vaccine preparation and antiviral therapy. In this paper, a weighted fragment conflict graph was constructed by introducing fuzzy distance, and a viral quasispecies haplotypes reconstruction algorithm CWSS was proposed based on color coding technology. Firstly, the CWSS algorithm preprocesses the fragment conflict graph in accordance with a given threshold. Secondly, under the condition that adjacent vertices must have different colors, all vertices of the graph are colored according to their sum of edge weigh and saturation value. Finally, quasispecies haplotypes are obtained by assembling the fragment with the same color. The time complexity of the CWSS algorithm is $O(m^2n+mn)$. Simulated sequencing fragment were adopted to compare the reconstruction performance and quality of the CWSS algorithm and the Dsatur one. The experimental results show that CWSS algorithm can obtain more accurate quasispecies and higher reconstruction performance than Dsatur algorithm.

Keywords Quasispecies, Haplotype, Weighted graph, Color coding, Fuzzy distance

1 引言

在生物医学领域,根据病毒的固定基因或蛋白来研制的病毒疫苗和抗体能有效防御和治疗艾滋病、流感等RNA病毒导致的疾病。RNA病毒在宿主细胞中呈现高复制率和高变异率,即病毒基因在复制过程中自我纠正能力差,每个复制周期中均有大量的碱基发生突变,很容易形成一系列基因组高度相似的变异体,即一些核酸序列结构高度相似的单体型,称这些单体型为病毒准种[1-3],其构成的群体成为病毒在宿主内的存在形式。在抗病毒药物和宿主自身免疫系统所带来的强大选择压力下,病毒种群中的准种快速演化,产生抗原性变异,导致疫苗效果减弱乃至失效[4]。因此,掌握病毒的基因结构和发展方向,重建病毒种群中不同准种单体型的基因序列,对推进病毒疫苗和药物的研究尤为必要,可为制定合理抗病毒治疗方案提供重要依据[5]。

由于技术的限制,直接通过生物手段获得病毒准种单体

到稿日期:2017-11-23 返修日期:2018-02-21 本文受国家自然科学基金项目(61363035,61762015,61502111,61662007),广西自然科学基金项目(2015GXNSFAA139288),"八桂学者"工程专项,广西多源信息挖掘与安全重点实验室系统性研究基金项目(14-A-03-02,15-A-03-02),广西科技基地和人才专项(AD16380008)资助。

黄 丹(1993-),女,硕士生,主要研究方向为生物信息学;吴璟莉(1978-),女,博士,教授,硕士生导师,CCF会员,主要研究方向为生物信息学、算法设计与分析,E-mail:wjlhappy@mailbox.gxnu.edu.cn(通信作者)。

型的费用和时间成本很高。高通量测序技术的快速发展,使得检测病毒种群中比例很低的准种成为可能。例如,对于人类免疫缺陷病毒(Human Immunodeficiency Virus,HIV)和乙型肝炎病毒(Hepatitis B Virus,HBV)等常见的病原性病毒基因组,利用 Illumina 测序仪可以测得 $10^4\sim10^5$ 的测序深度,从患者病毒分离物中获取数百万个长度为 100 个碱基对(basepair)的读长(read),以保证能够检测到种群中准种比率很低(如 0.1%)的单体型[6]。因此,利用病毒准种测序片段数据,通过计算手段重建准种单体型的计算问题应运而生并深受关注。

测序片段长度短、测序错误、突变位点少等因素使得病毒准种重建问题的求解非常困难。该问题主要分为错误校正、单体型重建和单体型频率估计 3 个子问题,本文主要针对单体型重建这一子问题进行研究。目前,研究者们大多基于图论的方法对该问题进行求解,取得了一些相关的研究成果。Eriksson 等[7]运用聚类方法对测序片段进行错误校正,然后基于片段图根据链分解重建单体型,再利用最大期望算法估算准种单体型的频率。Mancuso 等[8]基于扩增子构建片段图,通过最小二乘法求解叉平衡问题来解读图的顶点频率偏移。一旦图得到了平衡,则用最大带宽或贪婪算法解决准种谱重建问题。Zagordi 等[9]提出了求解方法 ShoRAH,该方法通过贝叶斯方法校正测序误差,利用图论重建单体型并估算频率[7],但该方法的重建精确率不高。Huang 等[10]提出了制定片段冲突图的方法,通过对图中顶点着色来求解准种的重建问题。文献[11]利用 Bron-Kerbosch 算法构造无向图,找出所有子图来重建病毒准种。Bu 等[6]构造了测序片段的无权冲突图,并运用贪心策略进行顶点着色(本文称其为 Dsatur 算法),获得了比 ShoRAH 软件[9]更好的重建效果。上述方法均基于无权图进行问题求解,忽略了片段之间的冲突程度等有效信息。针对该问题,本文引入一种衡量片段之间差异度的模糊距离,并构造带权的片段冲突图,提出了基于彩色编码技术的病毒准种单体型重建算法 CWSS(Coloring with Weight Sum and Saturation)。实验结果表明,基于带权冲突图的 CWSS 算法能够获得比 Dsatur 算法[6]更少的准种单体型,并且能够获得更高的重建精度。

2 问题及符号定义

病毒准种群内含有一组核苷酸结构高度相似的病毒准种单体型序列,每种单体型在群内有一个单体型比例(prevalence)。在这组序列上,存在一些单核苷酸变异(Single Nucleotide Variation,SNV)位点,在某位点上出现频率最高的基因称为主要等位基因(major allele),频率次高的基因称为次要等位基因(minor allele)。假设给定某病毒准种群内一组长度为 n 的单体型,经测序产生 m 条测序片段,记为片段矩阵 $Mat_{m\times n}$,其中每行 $Mat_i(i=1,\cdots,m)$ 代表第 i 条片段,每列代表一个核苷酸位点,元素 $mat_{ij}\in\{A,T,C,G,-\}(i=1,\cdots,m,j=1,\cdots,n)$(− 表示空值,即片段 i 未覆盖第 j 个位点,或片段 i 在第 j 个位点的取值未知)。

令 $X=\langle x_1,\cdots,x_n\rangle$ 和 $Y=\langle y_1,\cdots,y_n\rangle$ 为两条字符序列,海明距离 $HD(X,Y)$ 定义为 X 和 Y 中对应位置取值不同的位数,如式(1)所示:

$$HD(X,Y)=\sum_{j=1}^{n}d(x_j,y_j) \tag{1}$$

其中,

$$d(x_j,y_j)=\begin{cases}1, & \text{若 } x_j\neq-,y_j\neq-,\text{且 } x_j\neq y_j\\ 0, & \text{否则}\end{cases} \tag{2}$$

模糊距离 $FD(X,Y)$ 用于衡量字符序列 X 和 Y 的差异程度,其定义如式(3)所示:

$$FD(X,Y)=\frac{\sum_{j=1}^{n}x_j\otimes y_j}{|\{j|x_j\neq-\text{或 }y_j\neq-,j=1,\cdots,n\}|} \tag{3}$$

其中,

$$x_j\otimes y_j=\begin{cases}0, & \text{若 } x_j=y_j\\ 1, & \text{若 } x_j\neq y_j,\text{且 } x_j,y_j\neq-\\ 0.5, & \text{否则}\end{cases} \tag{4}$$

当 X 和 Y 表示两条片段时,$HD(X,Y)>0$ 意味着片段 X 和 Y 之间存在冲突,否则不存在冲突。若存在冲突,则表示其可能来自于不同的单体型,或存在测序错误。$FD(X,Y)$ 体现了片段 X 和 Y 的冲突程度,$FD(X,Y)$ 越大,则片段 X 和 Y 来自不同单体型的可能性越大。假定当所有片段均没有测序错误或错误已校正时,片段矩阵 Mat 中的行可被划分成一组只含有相容片段的子集,且每个子集中的片段确定一条单体型。病毒准种重建问题即给定片段矩阵 Mat,试图重建一组包含最少单体型的准种群,并为每个单体型估算比例。

3 CWSS 算法

本文提出一种基于彩色编码技术的病毒准种单体型重建算法 CWSS。算法输入为片段矩阵 Mat 和参数 p,输出为一组单体型 $\hat{Q}=(\hat{Q}_1,\hat{Q}_2,\cdots,\hat{Q}_K)$($K$ 的取值未知)。如图 1 所示,对于输入的片段矩阵 Mat,CWSS 算法首先对其进行错误校正和简化处理,即仅保留片段上的 SNV 位点;然后构造带权冲突图,并对冲突图进行顶点着色,将着相同颜色的顶点片段分别组装,得到一组结果单体型 $\hat{Q}'=(\hat{Q}_1',\hat{Q}_2',\cdots,\hat{Q}_K')$;最后将单体型 $\hat{Q}'$ 扩展为 $\hat{Q}=(\hat{Q}_1,\hat{Q}_2,\cdots,\hat{Q}_K)$。下面给出 CWSS 算法的详细步骤。

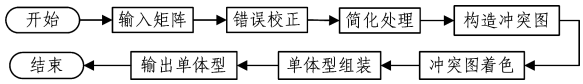

图 1 算法流程

Fig. 1 Algorithm flowchart

3.1 错误校正

虽然病毒准种群中比例很低(低于 1%)的个体上的 SNV 位点极易与测序错误(Illumina 测序仪的错误率低于 1%,实际为 0.1%~0.5%)发生混淆,但借助新一代测序技术[12]的超高测序深度可以将其有效地区分。假设某个准种的比例低于 1%,但对于某个特定的 SNV 位点,可由多个准种与其共享取值来提高此位点上共享取值的总体比例,通常认为若 SNV 位点上的次要等位基因取值比例大于 1%,则该位点是 SNV 位点,即变异位点;否则,该位点是非变异位点,其上所有与主要等位基因不同的取值均认定是测序错误。变异位点

上出现频率小于 1%的位点或在非变异位点存在测序错误的位点取值均设为空值－[13]。经过错误校正后,矩阵 Mat 上每个位点只有 3 种取值:主要等位基因取值、次要等位基因取值和空值－。为方便处理,用 0(1)表示主要等位基因(次要等位基因)取值,即 $mat_{ij}\in\{0,1,-\}(i=1,\cdots,m,j=1,\cdots,n)$。

3.2　简化处理

由于测序片段数量很大,为有效控制片段冲突图的规模,对片段矩阵进行如下 3 个方面的化简处理:1)删除冗余片段[6]。对于矩阵中任一片段 Mat_i,若存在片段 Mat_j 满足 $mat_{ik}=mat_{jk}(i,j=1,\cdots,m,i\neq j,k=1,\cdots,n)$,则删除片段 Mat_i。2)合并重叠片段[6]。若片段 Mat_i 和 $Mat_j(i,j=1,\cdots,m,i\neq j)$在其所有重叠位点$\{c_1,\cdots,c_k\mid 1\leqslant c_1,\cdots,c_k\leqslant n,mat_{k_l}\neq-,mat_{jc_l}\neq-,l=1,\cdots,k\}$上取值均相同,则将其合并,即 $mat_{il}=mat_{jl}(l=1,\cdots,n)$,删除片段 Mat_j。3)移除同值位点。由于取值相同的列对重建工作没有作用,若矩阵中片段在列 $j(j=1,\cdots,n)$上的取值全为 0(或 1),则将列 j 移除,并记该列的取值为 0(或 1)。为描述方便,化简后的新矩阵仍用 $Mat_{m\times n}$ 表示。

3.3　构造冲突图

利用化简后的矩阵 Mat 来构造片段冲突图 $G=\{V,E,W\}$,其中 $V=\{v_1,\cdots,v_m\}$为顶点集,$E=\{e_{ij}\mid e_{ij}=(v_i,v_j),v_i,v_j\in V\}$为边集,$W=\{w_{ij}\mid e_{ij}\in E\}$为边权集,其中 $w_{ij}=FD(Mat_i,Mat_j)$,顶点 v_i 对应矩阵中的片段 $Mat_i(i=1,\cdots,m)$,边 $e_{ij}\in E$ 表示片段 Mat_i 和 Mat_j 冲突,即 $HD(Mat_i,Mat_j)>0$ 且 $w_{ij}>p$。顶点 v_i 的边权和 $ws(i)$ 定义为 v_i 与其所有邻接顶点相连边上的权值和,如式(5)所示:

$$ws(i)=\sum_{e_{ij}\in E}w_{ij} \tag{5}$$

3.4　冲突图着色

本节给出基于贪心策略的启发式顶点着色方法,利用顶点的边权和及饱和度来选择着色点,试图运用最少的颜色数完成图 G 的顶点着色。为便于算法描述,给出如下符号定义:给定顶点 $v_i(i=1,\cdots,m)$,令 $cl(i)$ 表示其所着颜色,集合 $st(i)$ 记录顶点 v_i 所有邻接点的颜色,且 $|st(i)|$ 定义为 v_i 的饱和度,即其邻接顶点所具有的不同颜色数。为冲突图着色时,首先选定边权和最大的顶点进行着色,然后迭代选择饱和度最大的顶点进行着色,若迭代过程中顶点饱和度都相同,则选其边权和最大的顶点着色,直至所有顶点均被着色。算法详细步骤如算法 1 所示。

算法 1　顶点着色

输入:顶点未着色的片段冲突图 G

输出:顶点已着色的片段冲突图 G

```
1. for i=1,2,…,m
2.    st(i) =Ø;cl(i)=0;
3. i=arg max(ws(i));
          v_i∈V
4. cl(i)=1;
5. U=V-{v_i}; //U 记录未着色的顶点
6. for each v_j(e_ij∈E) //遍历 v_i 的邻居 v_j
7.    st(j) =st(j)∪{cl(i)};//更新 v_j 的邻接顶点颜色集
8. while (U≠Ø)
9. {
10. N_s={v_i|arg max|st(i)|}; //N_s 为最大饱和度顶点集
                v_i∈U
11. i=arg max(ws(i)); //选 N_s 中最大边权和的顶点
          i∈N_s
12. if (max(st(i))>|st(i)|) then //st(i)中最大颜色号大于 i 的饱和度
13.      cl(i)=min{k|1≤k≤max(st(i)),k∉st(i)};
14. else cl(i)=|st(i)|+1; //分配新的颜色号
15. for each v_j(e_ji∈E)
16.      if (cl(i)∉st(j)) then
17.         st(j) =st(j)∪{cl(i)};//更新 v_j 的邻接顶点颜色集
18.   U=V-{v_i};
19. }
```

3.5　单体型组装及其比例估算

组装单体型时,先对图中顶点 $v_i(i=1,\cdots,m)$按照其 $cl(i)$取值进行归类,即将 $cl(i)$取值为 k 的顶点片段归类到第 k 种颜色的片段集中($k=1,\cdots,K,K=\max(cl(i)),v_i\in V$)。将 K 种颜色集中所包含的片段拼接成 K 条单体$\hat{Q}'=(\hat{Q}_1',\hat{Q}_2',\cdots,\hat{Q}_K')$。然后,将简化处理时删除的列重新加回。对于重建单体型$\hat{Q}'=(\hat{Q}_1',\hat{Q}_2',\cdots,\hat{Q}_K')$,若某个已删除位点的取值记为 0(或 1),则在单体型 $\hat{Q}'$中各单体型相应缺失位点处插入 0(或 1),由此扩展成最终的单体型 $\hat{Q}=(\hat{Q}_1,\hat{Q}_2,\cdots,\hat{Q}_K)$。单体型$\hat{Q}_k$在重建准种中的比例($prevalence$)定义为着色为 k 的片段数($count$)占简化处理后的片段量(m)的百分比,如式(6)所示:

$$prevalence=\frac{count}{m}\times 100\% \tag{6}$$

3.6　算法复杂性

本节对 CWSS 算法的时间复杂性进行分析。算法分为 4 个阶段:1)错误校正阶段,处理片段矩阵的时间复杂度为 $O(mn)$;2)简化处理阶段,处理 $Mat_{m\times n}$ 片段矩阵的时间复杂度为 $O(mn)$,计算取值相同的列集的时间复杂度为 $O(mn)$;3)构造冲突图阶段以及冲突图着色阶段,冲突片段顶点连边和图中片段顶点着色的时间复杂度为 $O(m^2n)$;4)单体型组装以及扩展阶段的时间复杂度为 $O(mn)$。因此,算法总的时间复杂度为 $O(m^2n+mn)$。

4　实验结果

由于真实的片段数据很难得到,本文利用具有真实测序数据特征的模拟数据集[6]进行实验测试,对 CWSS 算法和 Dsatur 算法[6]的重建效果进行比较分析。实验在一台服务器(英特尔至强处理器 E5 2623 2.60 GHz,内存为 128 GB)上进行,操作系统为 Windows Server 2008 R2,编译运行工具为 Python2.7.13。

4.1　实验数据

本文采用了文献[6]提供的实验数据。数据选用人类免疫缺陷病毒Ⅰ型(HIV-1)HXB2 株 (GenBank Ko3455)env gp160 (基因中 6225-8792 碱基对所在位)作为参考序列来产生模拟准种,生成 $c=20$ 条相对于参考序列变异率为 2.5%的原始单体型基因序列,且各序列之间存在平均 4.7%的相异位点。

为仿真病毒群体中准种的实际比例，设置 20 条序列的比例分别约为 18.0%，13.0%，13.0%，10.0%，10.0%，8.0%，8.0%，5.0%，5.0%，3.0%，3.0%，1.0%，1.0%，0.6%，0.6%，0.3%，0.3%，0.1%，0.1%和 0.1%。针对模拟准种，分别根据 0%和 0.1%两种测序错误率来产生两组模拟测序片段数据，将其分别记为 A 组数据和 B 组数据。两组数据各含有 245852 条片段(122926 对，或 10000X 覆盖)，通过删除相同片段等数据处理后，A 组数据剩余 22479 条片段对，B 组数据剩余 39621 条片段对。

4.2 评价指标

本文用以下 7 个指标来评价算法的重建效果。

1)重建单体型种数(a)：根据测序片段重建的单体型个数。

2)成功重建的单体型种数(b)：一个原始单体型和一个重建单体型的序列比对一致率和比对长度均达到 80%以上时，称为有效比对。对于一个重建单体型，若有一个原始单体型是其有效比对，则称该个体为成功重建的个体。本文采用 Blastn 软件[14]进行序列比对。

3)精确率(precision)[13]：重建单体型中被成功重建的原始单体型比例，如式(7)所示：

$$precision=\frac{b}{a} \tag{7}$$

4)重召率(recallrate)[13]：原始单体型中被成功重建的比例，c 表示原始准种群中的单体型种数，如式(8)所示：

$$recallrate=\frac{b}{c} \tag{8}$$

5)F 值(F-measure)[13]：测试精度的度量，综合了考虑病毒准种单体型重建精确率和重召率，如式(9)所示：

$$F\text{-}measure=\frac{2(precision\times recallrate)}{precision+recallrate} \tag{9}$$

6)正确率(Correct Base，CB)：用于衡量原始准种 Q 与重建准种 $\hat{Q}$ 之间的吻合度，即重建准种 $\hat{Q}$ 正确构建单体型的核苷酸比例，如式(10)所示：

$$CB(Q,\hat{Q})=\frac{1}{c}\sum_{i=1}^{c}cb_i \tag{10}$$

其中，$cb_i=(1-\frac{HD(Q_i,\hat{Q}_k)}{n})\times 100\%$，$\hat{Q}_K(\hat{Q}_K\in\hat{Q})$ 为与 Q_i 差异最小的个体，即 $\hat{Q}_k=\arg\min_{\hat{Q}_j\in\hat{Q}}\{HD(Q_i,\hat{Q}_j)\mid k=1,\cdots,K\}$，$\hat{Q}_k$ 记为 Q_i 的重建个体。

7)单体型比例正确率(Prevalence Correct，PC)：原始准种群和重建准种群中对应单体型的比例(prevalence)符合度，采用皮尔逊相关系数来衡量，如式(11)所示：

$$PC=\frac{1}{c-1}\sum_{i=1}^{c}(\frac{x_i-\bar{x}}{\sigma(x)})(\frac{y_k-\bar{y}}{\sigma(y)}) \tag{11}$$

其中，x_i 和 y_k 分别为 $Q_i(i=1,\cdots,c)$ 和 $\hat{Q}_k(k=1,\cdots,K)$ 的比例($\hat{Q}_k$ 为 Q_i 的重建个体)；$\bar{x}$ 和 $\bar{y}$ 分别为对应比例均值；$\sigma(x)$ 和 $\sigma(y)$ 分别为对应比例的标准差。

4.3 结果与分析

本节对 CWSS 算法和 Dsatur 算法的性能进行测试对比分析。经过多组实验测试可知，当参数 p 设置为 0.4 时，算法 CWSS 重建效果最佳，下文实验中阈值 p 设置为 0.4。

表 1 给出 A 组数据的测试结果。如表 1 所列，两种算法在成功重建的单体型种数、重召率两个指标上具有相同的结果。相比于 Dsatur 算法，CWSS 算法在运行效率、重建单体型种数、精确率和 F 值指标上获得了更好的求解效果。例如，CWSS 算法的运行时间为 336 min，Dsatur 算法的运行时间为 570 min，前者比后者快约 0.76 倍；CWSS 重建出 20 种单体型，而 Dsatur 重建出 26 种单体型，前者比后者少 6 种；两种算法对应的精确率分别为 1 和0.8，F 值分别为 1 和 0.9。虽然 CWSS 算法在正确率和单体型比例正确率两个指标上略低于 Dsatur 算法，但仍具有较高的正确率，且重建准种群的单体型比例与原始准种群的单体型比例具有较高的匹配度。

表 1 A 组数据的实验比较结果

Table 1 Experimental comparison results of group A

参数	原始准种	Dsatur		CWSS	
运行时间/min		570		336	
原始单体型种数(c)	20	20		20	
重建单体型种数(a)	20	26		20	
成功重建的单体型种数(b)	20	20		20	
精确率(b/a)	1	0.8		1	
重召率(b/c)	1	1		1	
F 值	1	0.9		1	
正确率(CB)/%	100.0	99.7		98.4	
单体型比例正确率(PC)	1	0.8		0.7	
cb&. 比例	比例	cb	比例	cb	比例
单体型 1/%	18.0	100.0	7.6	98.9	5.6
单体型 2/%	13.0	100.0	8.1	98.4	7.9
单体型 3/%	13.0	99.5	6.2	96.7	7.9
单体型 4/%	10.0	99.1	6.6	98.8	7.2
单体型 5/%	10.0	99.4	6.9	99.1	9.2
单体型 6/%	8.0	100.0	7.6	98.8	10.8
单体型 7/%	8.0	99.6	7.0	99.3	10.5
单体型 8/%	5.0	99.3	6.8	99.0	6.7
单体型 9/%	5.0	100.0	7.8	98.9	7.0
单体型 10/%	3.0	99.6	5.1	98.9	4.3
单体型 11/%	3.0	100.0	6.4	98.8	4.6
单体型 12/%	1.0	99.5	3.2	99.1	2.9
单体型 13/%	1.0	99.8	3.2	98.9	3.3
单体型 14/%	0.6	99.6	2.6	99.0	2.3
单体型 15/%	0.6	100.0	3.1	98.9	2.3
单体型 16/%	0.3	100.0	2.1	98.8	1.5
单体型 17/%	0.3	99.7	1.9	96.4	1.4
单体型 18/%	0.1	99.3	1.8	96.0	2.2
单体型 19/%	0.1	100.0	0.8	96.1	0.8
单体型 20/%	0.1	100.0	1.1	98.9	1.2

表 2 给出 B 组数据的测试结果。如表 2 所列，CWSS 算法和 Dsatur 算法在运行效率、重建单体型种数、精确率、正确率和单体型比例正确率等指标上与表 1 的结果类似。由于 CWSS 算法和 Dsatur 算法针对 B 组数据得到的重建单体型种数分别为 21 和 44，均大于原始准种种数，故此处忽略重召率和 F 值指标。从表 2 中可以看出，CWSS 算法在运行效率、重建单体型种数、精确率 3 个指标上具有明显优势，但正确率和单体型比例正确率等指标略低于 Dsatur 算法。此外，PC 指标值的明显下降说明了测序错误率的引入对重建准种群中各单体型的比例估算有较大的负面影响。

表 2　B 组数据的实验比较结果

Table 2　Experimental comparison results of group B

参数	原始准种	Dsatur		CWSS	
运行时间/min		3060		235	
原始单体型种数(c)	20	20		20	
重建单体型种数(a)	20	56		21	
成功重建的单体型种数(b)	20	44		21	
精确率(b/a)	1	0.8		1	
重召率(b/c)	1	—		—	
F 值	1	—		—	
正确率(CB)/%	100.0	99.0		98.1	
单体型比例正确率(PC)	1	0.5		0.4	
cb& 比例	比例	cb	比例	cb	比例
单体型 1/%	18.0	98.8	4.9	98.3	10.6
单体型 2/%	13.0	98.7	0.4	98.3	1.9
单体型 3/%	13.0	99.1	5.8	98.2	8.3
单体型 4/%	10.0	98.6	1.8	98.4	17.3
单体型 5/%	10.0	99.3	5.8	98.5	8.0
单体型 6/%	8.0	99.3	5.9	98.4	3.2
单体型 7/%	8.0	98.9	4.5	98.2	5.3
单体型 8/%	5.0	99.4	5.6	98.1	1.0
单体型 9/%	5.0	98.9	2.9	98.1	0.6
单体型 10/%	3.0	98.6	0.3	98.1	4.2
单体型 11/%	3.0	99.1	4.5	97.8	2.3
单体型 12/%	1.0	99.2	2.1	97.7	0.6
单体型 13/%	1.0	99.3	2.6	98.0	0.8
单体型 14/%	0.6	99.4	2.2	98.1	7.9
单体型 15/%	0.6	99.5	2.4	98.4	1.2
单体型 16/%	0.3	98.9	2.3	98.1	0.5
单体型 17/%	0.3	99.2	1.3	98.0	5.9
单体型 18/%	0.1	98.9	1.1	97.5	4.9
单体型 19/%	0.1	98.8	2.1	98.1	1.7
单体型 20/%	0.1	98.8	0.8	97.7	13.1

下面对实验结果进行分析。CWSS 算法的主要优势在于更加精确地重建出原始准种群。由于高通量测序产生大量的片段数据，由此构建的片段冲突图极其庞大。CWSS 算法利用冲突图中边权值差异度大的数据特点，通过设置合理阈值 p，去除相对权值较小的边，从而减少图中边缘化的片段顶点，提高片段间的集中度。在此基础上，CWSS 算法对冲突图中的片段顶点进行精准归类，并将这些片段集组装成单体型，以实现准种单体型的重建。由实验结果可知，CWSS 算法能够在更短的运行时间内获得更少的重建单体型种数和更高的精确率。

结束语　重建准种单体型对研究病毒准种中不同单体型的基因结果，从而研发流行性病毒的疫苗和制定合理的抗病毒治疗方案有重要的实际指导意义，准种重建是获得单体型数据的有效手段。本文针对带权冲突图模型，对病毒准种单体型重建问题进行研究，提出了基于彩色编码技术的准种重建算法 CWSS。CWSS 算法根据冲突片段顶点相应边权和与所在着色片段顶点邻接顶点的饱和度，来标记片段顶点并进行着色，算法所得颜色数即为重建种数。实验结果显示，CWSS 算法能获得更少的重建种数和更高的精确率，具有较好的实用价值。该彩色编码技术还可扩展用于更高序列错误的病毒准种单体型重建，但由于数据量大导致内存溢出，相应的实验对设备性能的要求更高，目前未能获得实验结果。故今后将对其展开进一步的研究。

参 考 文 献

[1] EIGEN M，SCHUSTER P. Hypercycle-principle of natural self-organization. a. emergence of hypercycle [J]. Naturwissenschaften，1977，64(11)：541-565.

[2] ANDINO R，DOMINGO E. Viral quasispecies[J]. Virology，2015，479-480(1)：46-51.

[3] DOMINGO E，SHELDON J，PERALES C. Viral quasispecies evolution [J]. Microbiology & Molecular Biology Reviews Mmbr，2012，76(2)：159-216.

[4] COX R J，BROKSTAD K A，OGRA P. Influ-enza virus：immunity and vaccination strategies. Comparison of the immune response to inactivated and live，attenuated influenza vaccines[J]. Scandinavian Journal of Immunology，2004，59(1)：1-15.

[5] LAURING A S，ANDINO R. Quasispecies theory and the behavior of RNA viruses [J]. Plos Pathogens，2010，6(7)：e1001005.

[6] BU D，TANG H. Quasispecies reconstruction based on vertex coloring algorithms [C] // IEEE International Conference on Bioinformatics and Biomedicine. Washington D. C：IEEE Computer Society，2014：63-66.

[7] ERIKSSON N，PACHTER L，MITSUYA Y，et al. Viral population estimation using pyrosequencing[J]. Plos Computational Biology，2008，4(5)：e1000074.

[8] MANCUSO N，TORK B，SKUMS P，et al. Viral Quasispecies Reconstruction from Amplicon 454 Pyrosequencing Reads[C]// IEEE International Conference on Bioinformatics and Biomedicine Workshops. Washington D. C：IEEE Computer Society，2011：94-101.

[9] ZAGORDI O，BHATTACHARYA A，ERIKSSON N，et al. ShoRAH：estimating the genetic diversity of a mixed sample from next-generation sequencing data[J]. Bmc Bioinformatics，2011，12(1)：1-5.

[10] HUANG A，KANTOR R，DELONG A，et al. QColors：An algorithm for conservative viral quasispecies reconstruction from short and non-contiguous next generation sequencing reads[J]. Silico Biology，2011，11(5-6)：193-201.

[11] BRON C，KERBOSCH J. Algorithm 457：finding all cliques of an undirected graph [J]. Communications of the Acm，1973，16(9)：575-576.

[12] METZKER M L. Sequencing technologies－the next generation [J]. Nature Review Genetic，2010，11(1)：31-46.

[13] POH W T，XIA E，CHININMANU K，et al. Viral quasispecies inference from 454 pyrose-quencing[J]. BMC Bioinformatics，2013，14(1)：355.

[14] ALTSCHUL S F，GISH W，MILLER W，et al. Basic local alignment search tool [J]. Journal of Molecular Biology，1990，215(3)：403-410.