基于 Contig 的单面基因组框架填充 2-近似算法

柳 楠,卞忠勇,李 洋,朱永琦
(山东建筑大学 计算机科学与技术学院,山东 济南 250101)

摘 要:随着基因测序技术的持续发展,基因组框架填充问题受到广泛关注。该文针对基于 contig 的单面含重复基因的基因组框架填充问题开展研究。通过设计有效的近似算法,完成根据参照基因组,将缺失基因填充至基因测序获得的不完整框架中,提高基因组框架的完整性。前期研究的基因组框架填充问题,缺失基因可以插入到不完整序列的任意两个基因之间,而基于片段重叠群(contig)的基因组框架填充,缺失基因的插入位置被限制在两个 contig 之间,更具一般性,该问题已被证明是 NP 完全问题。现有的近似算法中,2-近似算法处理的实例具有特殊性,2.57-近似算法针对一般实例,但近似性能比不够理想。该文以缺失基因、基因位点和断点三者之间的对应关系为基础,采用贪婪策略和最大匹配相结合的方式避免在填充过程中出现冗余公共邻接,并通过生成新的 contig 增加外邻接的数量,将针对一般实例的算法近似性能比提高到 2,完成了基于 Python 的可视化程序开发,进一步验证了算法的有效性。

关键词:基因组;框架填充;近似算法;贪婪策略;最大匹配
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2024)02-0148-08
doi:10.3969/j.issn.1673-629X.2024.02.022

A 2-Approximation Algorithm for Contig-based One-sided Genome Scaffold Filling

LIU Nan, BIAN Zhong-yong, LI Yang, ZHU Yong-qi
(School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China)

Abstract: The problem of genome scaffold filling gets more and more attention with the advancement of genome sequencing technology. The contig-based one-sided genomic scaffold filling problem is researched. Effective approximation algorithms are devised to enhance the integrity of the genome scaffold. The objective of algorithms is to insert the missing genes derived from genome sequencing into the scaffold according to the reference genome. In previous research on genome scaffold filling, missing genes can be inserted between any two genes in an incomplete sequence. While in genome scaffold based on contigs, the insertion position of missing genes is limited to between two contigs. This problem has been proven to be NP-complete and more general. Two related algorithms are analyzed. The 2-approximation is not for general but special case, and the approximation performance ratio of the 2.57-approximation algorithm is considered unsatisfactory. A new algorithm focusing on the correspondence between missing genes, gene slots and breakpoints is designed by using greedy method and maximal matching. The algorithm not only solves the redundant common adjacencies problem, but also increases the number of external adjacencies by generating new contigs. As a result, the approximation performance for general case is improved to a ratio of 2. The effectiveness of the algorithm is further verified by developing a visualization program based on Python.

Key words: genome; scaffold filling; approximation algorithm; greedy strategy; maximum matching

0 引 言

近年来,为了高效获得完整的生物基因组序列,基因测序和组装技术的发展受到广泛关注[1-2]。很多时候,使用生物测序手段无法直接获得完整的基因组序列,通过使用计算机技术将多次测序获取的基因组框架进行组装,将缺失基因填充到基因组框架,能够提高基因组序列的完整性[3-4]。

2010 年,Muñoz 等人提出了基因组框架填充问题[5],给定完整基因组 R 和待填充框架 I,规定 R 和 I 均不含重复基因,将缺失基因全部插入到框架 I 中得到 I',使 I' 与 R 之间的基因重组距离(DCJ 距离)最小[6],Muñoz 等人证明了这个问题是多项式时间可解

收稿日期:2023-05-18 **修回日期**:2023-09-20
基金项目:国家自然科学基金项目(61902221);山东省自然科学基金项目(ZR2018MF012)
作者简介:柳 楠(1980-),女(满族),教授,博士,CCF 会员(H4774M),研究方向为算法分析与设计、复杂性理论和生物计算;通信作者:卞忠勇(1997-),男,硕士研究生,研究方向为算法分析与设计、计算基因组学。

的。为使基因组框架填充问题更接近于现实情况,接下来许多研究的问题定义中,基因序列包含了重复基因,这使得缺失基因插入位置的选择变得复杂[7]。在基于公共邻接的含重复基因的基因组框架填充问题中,给定含重复基因的完整基因组 G 和待填充框架 S,使填充缺失基因后获得的 S' 与 G 之间公共邻接数最大[8],Jiang Haitao 等人证明该问题是 NP 完全问题,并提出 1.33-近似算法[9-10]。

前面两问题中基因组框架的基本单元都是单个基因,对缺失基因插入位置不做限制。但实际应用中,框架多由一系列片段重叠群(contig)组成,缺失基因只允许在 contig 的两端插入[11]。对于基于 contig 的单面框架填充的研究[12],Jiang Haitao 等人通过将哈密顿路径问题多项式变换到本问题,证明它是 NP 完全问题,然后使用贪婪策略和最大匹配提出 2-近似算法[13],Tan Guanlan 等人指出这个算法不具一般性,并通过构造辅助图和最大匹配设计验证了 2.57-近似算法[14]。Bulteau L 等人基于最大邻接数、最小断点距离提出了 k-Mer 参数的 FPT 算法[15],Ma Jingjing 等人提出以 duo-preservations 作为度量,设计实现了近似比为 2 的一个基本算法,并通过贪婪策略将近似比提高到 1.71[16]。

该文研究的问题是基于 contig 的单面含重复基因组框架填充问题,研究发现该问题的 2-近似算法不能避免出现冗余公共邻接,使近似性能比无法达到 2,文献[14]提出了一种解决方案,但近似性能比只能达到 2.57。该文通过构造以缺失基因、基因位点、断点为节点的辅助图,配合最大匹配、贪婪策略,设计了一个近似算法,证明了算法的近似性能比可达到 2,并实现了算法的验证程序。

1　相关定义

给定集合 Σ,Σ 包含所有用于标识基因的字符。设 A 是由 Σ 中元素构成的字符串,A 中每个元素可多次出现,称 A 为序列。用 $M(A)$ 表示 A 中的字符集合,由于 $M(A)$ 可能是多重集,为方便计算,用 $NM(A)$ 记录每个字符的重数。例如 $\Sigma = \{a,b,1,2\}$,$A = ab12a2$,$M(A) = \{a,a,b,1,2,2\}$,$NM(A) = \{a:2,b:1,1:1,2:2\}$。若 x 和 y 在 A 中相邻,则称 xy 为 A 中的一基因对,用 $P(A)$ 表示 A 中所有基因对的集合,$P(A)$ 也可能是多重集,用 $NP(A)$ 记录基因对重数。前面例子中 $P(A) = \{ab,b1,12,2a,a2\}$,$NP(A) = \{ab:1,b1:1,12:1,2a:2\}$。易知基因对 $2a$ 与 $a2$ 无区别,故后面讨论中 xy 与 yx 视为相同基因对。

给定两条序列 $A = a_1a_2\cdots a_m$ 和 $B = b_1b_2\cdots b_n$,若 $P(A)$ 中的一基因对 a_ia_{i+1},在 $P(B)$ 可以找到 b_kb_{k+1} 与之对应($a_ia_{i+1} = b_kb_{k+1}$ 或 $a_ia_{i+1} = b_{k+1}b_k$),称 a_ia_{i+1} 和 b_kb_{k+1} 是公共邻接,用 $a(A,B)$ 记录 A 和 B 全部的公共邻接,易知 $D_A(A,B) = P(A) - a(A,B)$ 和 $D_B(A,B) = P(B) - a(A,B)$ 分别表示 A 和 B 中没有匹配到公共邻接的基因对,这些基因对称为 A 或 B 中的断点。示例如图 1 所示。

$$A = c2a2cbde$$
$$B = c2ca2de$$
$$P(A) = \{c2,2a,a2,2c,cb,bd,de\}$$
$$P(B) = \{c2,2c,ca,a2,2d,de\}$$
$$a(A,B) = \{c2,2a,2c,de\}$$
$$D_A(A,B) = \{a2,cb,bd\}$$
$$D_B(A,B) = \{ca,2d\}$$

图 1　公共邻接和断点的示例

在基于 contig 的情况下,两条序列中一条是完整的基因序列 G,另一条是待填充的基因框架 $S = \bullet[C_1]\bullet[C_2]\bullet\cdots\bullet[C_n]\bullet$,$S$ 由多个 contig 序列 C_i 组成,contig 两边的 $\bullet$ 为可插入的基因位点,称为 slot。

下面正式给出基于 contig 的单面基因组框架填充问题(contig-based one-sided scaffold filling, Contig-One-Sided-SF-max)的定义。

定义 1:Contig-One-Sided-SF-max。

输入:一条含重复基因的完整基因序列 $G = g_1g_2\cdots g_m$ 和一条缺失部分基因的框架 $S = \bullet[C_1]\bullet[C_2]\bullet\cdots\bullet[C_n]\bullet$,其中 g_i 为单个基因,C_i 是 contig,且 $P(S) \subseteq P(G)$,$m > n$。设 $X = M(G) - M(S)$ 为 S 中的缺失基因集合,$X \neq \varnothing$。

问题:将 X 中的基因插入 S 中的 slot 得到 S',使得 $|a(G,S')|$ 最大。

输出:填充后的完整序列 S'。

2　Contig-One-Sided-SF-max 近似算法

给定实例 (G,S),研究目标是设计算法将缺失基因插入 S 得到 S',实现公共邻接数 $|a(G,S')|$ 最大化。初始公共邻接数 $a(G,S)$ 由已知可直接计算,故算法关注的重点是 $|a(G,S') - a(G,S)|$,即新增公共邻接数量。另外注意到 $a(G,S') - a(G,S) \subseteq D_G(G,S)$,因此在算法中判断基因插入位置需要以 G 中的断点集合 $D_G(G,S)$(简称 D)为依据,用 ND 记录 D 中每个断点的重数。新产生的公共邻接与 D 中的元素需一一对应,避免出现冗余公共邻接。

新产生的公共邻接根据缺失基因的插入位置的不同可以分为两类:

(1)外邻接:$C_i = c_{i\,\mathrm{start}}\cdots c_{i\,\mathrm{end}}$ 是 S 中的 contig,x 属于 S 的缺失基因集合 X,若 xy 可以构成公共邻接,且 $y = c_{i\,\mathrm{start}}$ 或 $y = c_{i\,\mathrm{end}}$,则称 xy 为外邻接。

(2)内邻接:除外邻接以外的其他新增公共邻接称为内邻接。

缺失基因串根据其插入 S 后所产生的公共邻接数量可以分为三类:

n -type Ⅰ串:由 n 个缺失基因组成的字符串(记做 Len- n 串)插入 S 中,新增 $n+1$ 个公共邻接,将该基因串称为 n -type Ⅰ串。显然 n -type Ⅰ串只能插入形如 $c_{i\,\mathrm{end}}]\bullet[c_{i+1\,\mathrm{start}}$ 的位点,该类位点记为 slot Ⅰ。

n -type Ⅱ串:Len- n 串插入 S 中,共新增 n 个公共邻接,就将该基因串称为 n -type Ⅱ串。显然 n -type Ⅱ串只能插入形如 $\bullet[c_{i\,\mathrm{start}}$ 或 $c_{i\,\mathrm{end}}]\bullet$ 的位点,该类位点记为 slot Ⅱ。

n -type Ⅲ串:Len- n 串插入 S 中,共新增 $n-1$ 个公共邻接,就将该基因串称为 n -type Ⅲ串。

如图 2 所示,Len-5 串 $deegg$ 插入 S,新增 cd, de, ee,eg,gg,$g4$ 六个公共邻接,故 $deegg$ 是 5-type Ⅰ串, $c]\bullet[4$ 为 slot Ⅰ型位点。Len-1 串 a,a,b 插入 S,分别得到 $a2$,$a3$,ab 各一个公共邻接, $a2$ 和 $2a$ 记为 $a2$,故 a,a,b 都是 1-type Ⅱ串, $\bullet[a,2]\bullet,3]\bullet$ 都是 slot Ⅱ型位点。在新增公共邻接中 $a2$,$a3$,ab,cd,$g4$ 为外邻接, de,ee,eg,gg 为内邻接。

$$G=a2a3abcdeegg4d161b5b$$
$$S=\bullet[adb]\bullet[bc]\bullet[452]\bullet[6113]\bullet$$
$$X=\{a,a,b,d,e,e,g,g\}$$
$$D=\{a2,a2,a3,a3,ab,b1,b5,b5,cd,de,d1,d4,ee,eg,gg,g4,61\}$$
$$ND=\{a2:2,a3:2,ab:1,b1:1,b5:2,cd:1,de:1,d1:1,d4:1,ee:1,eg:1,gg:1,g4:1,61:1\}$$
$$\text{最优解}S^{*}=\bullet b[adb]\bullet[bc]deegg[452]a\bullet[6113]a\bullet$$

图 2 Contig-One-Sided-SF-max 问题示例

2.1 算法核心思想框架

为方便理解,算法的大致轮廓如图 3 所示。

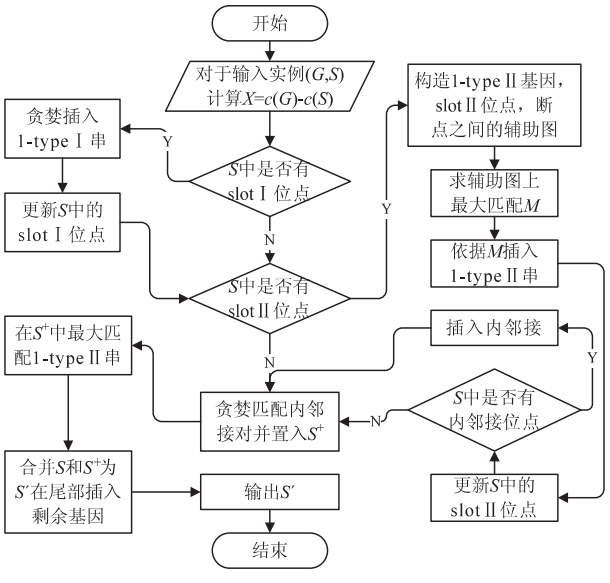

图 3 算法核心思想

2.2 算法初始化

对实例 (G,S) 进行初始化,得到 S 中的缺失基因集合 X 和断点集合 D,具体算法如算法 1 所示。

算法 1:Init (G,S)。

输入:完整基因序列 G,基因框架 S

输出:缺失基因集合 X,断点集合 D

1 $X\leftarrow\varnothing$, $D\leftarrow\varnothing$;
2 for $a\in M(G)$ do
3 if $NM(G)[a]>NM(S)[a]$ and $a\notin$ X then
4 添加 $NM(G)[a]-NM(S)[a]$ 个 a to X;
5 for $ab\in P(G)$ do
6 if $NP(G)[ab]>NP(S)[ab]$ and $ab\notin D$ then
7 添加 $NP(G)[ab]-NP(S)[ab]$ 个 ab to D;
8 return X and D

2.3 算法实现

下面对算法每一步做出详细介绍:

步骤一:采用贪婪策略插入 1-type Ⅰ串。

对于 X 中的元素 a,从左向右扫描框架 S,发现有 a 可插入的 slot Ⅰ型位点,便将 a 插入 S,将 a 从 X 中移除一个,更新 S 中的基因位点并将新增的两个公共邻接从 D 中删除。具体算法如算法 2 所示。

算法 2:Greedy-one(X, D, S)。

输入:缺失基因集合 X,断点集合 D,基因框架 $S=\bullet[C_1]\bullet[C_2]\bullet\cdots\bullet[C_n]\bullet$

输出:更新后的缺失基因集合 X,断点集合 D,基因框架 S

1 for $a\in X$ do
2 for $C_i=c_{i\,\mathrm{start}}\cdots c_{i\,\mathrm{end}}\in S$ do
3 if $i<n$ and $\{c_{i\,\mathrm{end}}a,ac_{i+1\,\mathrm{start}}\}\subseteq D$ and
4 ($c_{i\,\mathrm{end}}\neq c_{i+1\,\mathrm{start}}$ or ($c_{i\,\mathrm{end}}=c_{i+1\,\mathrm{start}}$ and
5 $ND[c_{i\,\mathrm{end}}a]\geqslant 2$)) then $X\leftarrow X-\{a\}$,
6 $C_i\leftarrow c_{i\,\mathrm{start}}\cdots c_{i\,\mathrm{end}}a$, $C_i=C_i\cup C_{i+1}$,
7 $D\leftarrow D-\{c_{i\,\mathrm{end}}a,ac_{i+1\,\mathrm{start}}\}$,
8 $ND[c_{i\,\mathrm{end}}a]\leftarrow ND[c_{i\,\mathrm{end}}a]-1$,
9 $ND[ac_{i+1\,\mathrm{start}}]\leftarrow ND[ac_{i+1\,\mathrm{start}}]-1$;
10 return X and D and S

对于图 2 的例子,这一步的操作是将 d 插入到 $c]\bullet[4$ 中,与 2-近似算法结果一致。

步骤二:采用最大匹配插入 1-type Ⅱ串。

设 $S\text{Ⅱ}=\{s_1,s_2,\cdots,s_{2n}\}$ 为 S 中 slot Ⅱ位点集合, n 等于 S 中 contig 的数量, $S\text{Ⅱ}_e=\{e_1,e_2,\cdots,e_{2n}\}$, e_i 为位点 s_i 处相邻的基因,构造集合 X, D, SⅡ之间辅助图 Γ, E_{XS} 记录 X 中元素与 SⅡ中元素之间的边, E_{SD} 记录 SⅡ中元素与 D 中元素之间的边。

构造 Γ 后,从中计算出匹配结果 M_1, M_1 是多个三元组 (a,s_i,xy) 的集合,根据 M_1 来插入 1-type Ⅱ串和更新断点集合 D。具体算法如算法 3 所示。

算法 3:Max-match(X, D, S, $S\text{Ⅱ}$, $S\text{Ⅱ}_e$)。

输入:缺失基因集合 X,断点集合 D,基因框架 S,slot Ⅱ位

点集合 $S\text{Ⅱ}$, $S\text{Ⅱ}_e$

输出:更新后的缺失基因集合 X ,断点集合 D ,基因框架 S

$E_{XS} \leftarrow \varnothing$, $E_{SD} \leftarrow \varnothing$, $V \leftarrow \varnothing$, $E \leftarrow \varnothing$;
newS Ⅱ $\leftarrow \varnothing$, newS Ⅱ$_e \leftarrow \varnothing$;
$D' \leftarrow D$ 中由 X 和 $S\text{Ⅱ}_e$ 元素组成的断点;
for $a \in X$ do
 for $s_i \in S\text{Ⅱ}$ do
  if $as_i \in D'$ then $E_{XS}[a][s_i] \leftarrow 1$;
  else $E_{XS}[a][s_i] \leftarrow 0$;
for $s_i \in S\text{Ⅱ}$ do
 for $xy \in D'$ do
  if $s_i = x$ and $y \in$ X and $E_{XS}[y][s_i] = 1$
   then $E_{SD}[y][s_i][xy] \leftarrow 1$;
   else $E_{SD}[y][s_i][xy] \leftarrow 0$;
 $V \leftarrow S \cup S\text{Ⅱ} \cup D'$, $E \leftarrow E_{XS} \cup E_{SD}$;
 求出 $\Gamma(V,E)$ 上匹配结果 M ;
 for $M[i] \in M$ do
  将 $M[i][0]$ 插入到 S 中的 $M[i][1]$ 位点,
  newS Ⅱ $\leftarrow$ newS Ⅱ $+ M[i][1]$,
  newS Ⅱ$_e \leftarrow$ newS Ⅱ$_e + M[i][0]$,
  $X \leftarrow X - \{M[i][0]\}$, $D \leftarrow D - \{M[i][2]\}$;
 return X and D and S

对于图 2 的例子,2-近似算法在这一步是将缺失基因 a,a,b 与 slot Ⅱ 位点做最大匹配,如图 4 所示,经过二分匹配后将 a 插入到 $b]\bullet$,将另一个 a 插入到 $\bullet[b$, b 插入到 $\bullet[a$,三个基因插入后得到三个基因对 ab ,而断点集合中只有一个断点 ab ,故经过此步操作只能得到一个新增公共邻接 ab 。

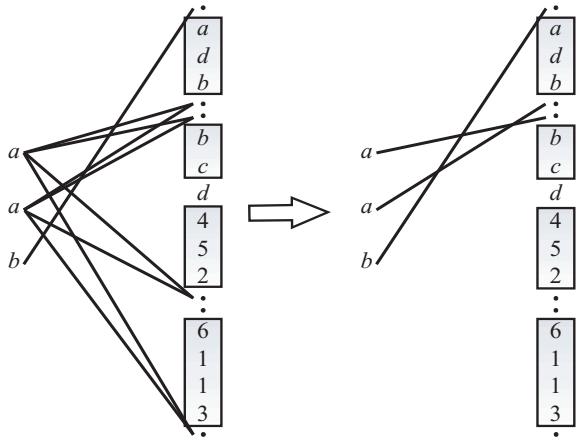

图 4　基于缺失基因、基因位点的匹配算法

文中算法在这步将缺失基因 a,a,b 与 slot Ⅱ 位点、断点 ab , $3a$, $2a$ 之间做最大匹配,如图 5 所示,匹配结果 $M_1 = \{(a,s_6,3a),(a,s_4,2a),(b,s_1,ab)\}$,根据 M_1,将 a 插入到 $3]\bullet$,另一个 a 插入到 $2]\bullet$, b 插入到 $\bullet[a$,这样便得到 $2a$, $3a$, ab 三个新增公共邻接。

步骤三:采用最大匹配,在上一步更新的位点上插入内邻接。

与步骤二相似,这一步仍在缺失基因、位点和断点之间做最大匹配, $S\text{Ⅱ}$ 只取上一步更新得到的位点即可,调用 Max-match(X , D , S , newS Ⅱ , newS Ⅱ$_e$)算法完成内邻接的插入。

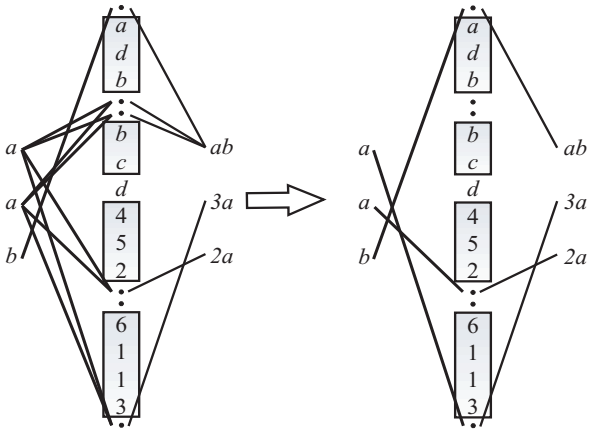

图 5　基于缺失基因、基因位点、断点的匹配算法

对于图 2 的例子, $X = \{e,e,g,g\}$, $D = \{a2,a2,a3,a3,ab,b1,b5,b5,cd,de,d1,d4,ee,eg,gg,g4,61\}$,步骤二中更新的位点为 $\bullet[b$, $a]\bullet$, $a]\bullet$,显然 D 中与缺失基因存在边的所有断点都不能和这些位点构成边,无法构图进行匹配,故跳过此步。

步骤四:采用贪婪策略,在 X 中匹配内邻接对,将获得的内邻接对作为新 contig 置入集合 S^+ 。

对于 X 中的元素 a ,依次扫描其后面的其他元素,若有元素 b ,满足 ab 是 D 中的断点,则将 ab 作为长度为 2 的新 contig 置于 S^+ 中,并从 D 中移除一个断点 ab 。具体算法如算法 4 所示。

算法 4:Greedy-two(X , D)。

输入:缺失基因集合 X ,断点集合 D

输出:更新后的缺失基因集合 X ,断点集合 D ,新 contig 集合 S^+

$i \leftarrow 1$, $S^+ \leftarrow \varnothing$;
for $a \in X$ do
 for $b \in X - \{a\}$ do
  if $ab \in D$ then
   $C_i \leftarrow ab$, $S^+ \leftarrow S^+ \cup C_i$,
   $i \leftarrow i + 1$, $D \leftarrow D - \{ab\}$;
return X and D and S^+

对于图 2 的例子 $X = \{e,e,g,g\}$,实际计算中 X 中元素顺序并不确定,故经过步骤四会有两种可能:

(1)先得到内邻接 ee ,然后又得到内邻接 gg ,所以 $S^+ = \bullet[ee]\bullet[gg]\bullet$ 。

(2)先得到内邻接 eg ,剩下缺失基因只有 e 和 g ,但是 D 中已经没有断点 eg ,所以 $S^+ = \bullet[eg]\bullet$ 。

步骤五:采用最大匹配在 S^+ 中插入 1-type Ⅱ 串。

调用算法 Max-match(X , D , S^+ , S^+ Ⅱ , S^+ Ⅱ$_e$)得到填充了 1-type Ⅱ 串的 S^+ ,下面对步骤四中两种可能的结果继续进行讨论:

(1)若步骤四中得到的 $S^+ = \bullet[ee]\bullet[gg]\bullet$,则 X 为空,此步无任何操作。

(2)若步骤四中得到的 $S^+ = \bullet[eg]\bullet$,调用算法 Max-match($X, D, S^+, S^+ \text{ Ⅱ}, S^+ \text{ Ⅱ}_e$),得到的匹配结果为 $M_2 = \{(e, s_1, ee), (g, s_2, gg)\}$,填充后 $S^+ = \bullet e[eg]g\bullet$ 。

步骤六:合并 S 和 S^+ 为 S' , X 中的剩余基因,在不破坏现有邻接情况下,可插入 S' 中任意位点。

步骤七:输出 S' 。

可以将步骤三~步骤五合并称为匹配内邻接。在 2-近似算法中也有相关操作,该算法是将插入 1-type Ⅰ串后的剩余所有基因(包括二分匹配后插入的基因)构造多重图,在多重图上计算最大匹配,对于图 2 中的例子,以 $\{a, a, b, e, e, g, g\}$ 为顶点构造多图,断点有 $\{a2, a3, b1, b5, b5, de, d1, ee, eg, gg, g4, 61\}$ 。

如图 6 所示,通过构造多重图获得匹配结果是两对基因对 eg 和 eg ,而断点中只有一个 eg ,所以 2-近似算法匹配内邻接只新增一个公共邻接,最终得到 $S' = \bullet b[adb]a\bullet a[bc]d[452]a\bullet[6113]a\bullet eg\bullet eg\bullet$,总计新增 ab , cd , $d4$, eg 共 4 个公共邻接。在最优解 $S^* = \bullet b[adb]\bullet[bc]deegg[452]a\bullet[6113]a\bullet$ 中共有 9 个新增公共邻接,近似比为 9/4,出现这种情况的原因就是在最大匹配内邻接和外邻接时,都出现了冗余公共邻接现象,从而影响了该算法的近似比。

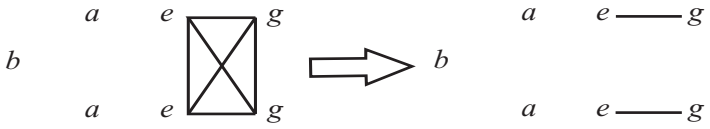

图 6 2-近似算法中多重图最大匹配

对于图 2 的例子,文中算法结果 $S' = \bullet b[adb]\bullet[bc]d[452]a\bullet[6113]a\bullet[ee]\bullet[gg]\bullet$ 或 $S' = \bullet b[adb]\bullet[bc]d[452]a\bullet[6113]a\bullet e[eg]g\bullet$,分别新增 7 个和 8 个公共邻接,近似比小于 2。

2.57-近似算法同样考虑到了避免冗余公共邻接的问题,通过在缺失基因和 contig 端点基因之间构造辅助图的方式消除冗余边,求出最大匹配来填充基因。但是该算法对于匹配结果与缺失基因和位点之间的对应关系没有做出判断。对于图 2 的例子,依据 2.57-近似算法,可以得到最大匹配结果中的 (a, b) 这样一组匹配,但是依据 (a, b) 是将 a 插入 $\bullet[b$,还是将 b 插入 $a]\bullet$ 并没有明断判断,若是前者将导致 b 无法插入框架,若是后者则 a 和 b 都可以插入。由此可见 2.57-近似算法虽然做到了避免冗余公共邻接,但是对于外邻接处理不佳,导致近似比大于 2。

2.4 算法的性质

在本节,通过对比最优解中和文中算法所求近似解中各类型串新增公共邻接数,证明算法近似性能比为 2。设 OPT 为最优解中的新增公共邻接数, $\text{OPT}_\text{Ⅰ}$, $\text{OPT}_\text{Ⅱ}$, $\text{OPT}_\text{Ⅲ}$ 分别为最优解中由 type Ⅰ, type Ⅱ, type Ⅲ类型基因串新增公共邻接数,可得:

$$\text{OPT} = \text{OPT}_\text{Ⅰ} + \text{OPT}_\text{Ⅱ} + \text{OPT}_\text{Ⅲ} \quad (1)$$

设 APP 为文中算法所求近似解中新增公共邻接数, $\text{APP}_\text{Ⅰ}$, $\text{APP}_\text{Ⅱ}$, $\text{APP}_\text{Ⅲ}$ 分别为近似解中由 type Ⅰ, type Ⅱ, type Ⅲ类型基因串新增公共邻接数,可得:

$$\text{APP} = \text{APP}_\text{Ⅰ} + \text{APP}_\text{Ⅱ} + \text{APP}_\text{Ⅲ} \quad (2)$$

文中提到的“新增公共邻接比”指一条或多条基因串在最优解中新增公共邻接数与在文中算法中新增公共邻接数之比。设某最优解中的 Len-k 串经过步骤四、五得到的内邻接数为 $nl(k)$ 。

引理 1 $nl(k) \geqslant \frac{k}{2}$

证明:假设 k 个缺失基因在最优解中可以组成长度为 k 的连续基因串,则这些缺失基因通过最大匹配可得到 $\lfloor k/2 \rfloor$ 个匹配,即 $\lfloor k/2 \rfloor$ 个内邻接。在步骤四中,使用的是贪婪策略,在最坏情况下,最大匹配中两个匹配对错位匹配,会使得两个基因无法匹配,如:断点有 ac , ab , cd ,缺失基因有 a , b , c , d ,最大匹配为 $\{ab, cd\}$,若 ac 匹配到一起,则 b , d 无法匹配,称 ac 为错位匹配。

因此,在步骤四使用贪婪策略可得到 m 个内邻接, m 的取值范围是:

$$\frac{1}{2}\lfloor \frac{k}{2} \rfloor \leqslant m \leqslant \lfloor \frac{k}{2} \rfloor$$

$m = \frac{1}{2}\lfloor \frac{k}{2} \rfloor$ 时,在步骤五每个新 contig 两端均可以新增公共邻接,所以在步骤四、五总计得到的内邻接数为:

$$nl(k) = \frac{1}{2}\lfloor \frac{k}{2} \rfloor * (1 + 2) = \frac{3}{2}\lfloor \frac{k}{2} \rfloor$$

用数学归纳法证明 $\frac{3}{2}\lfloor \frac{k}{2} \rfloor \geqslant \frac{k}{2}$:

$k = 1$ 和 2 时显然成立。

设 $k = l$ ($l > 2$)时成立,即 $\frac{3}{2}\lfloor \frac{l}{2} \rfloor \geqslant \frac{l}{2}$,则 $k = l + 1$ 时, $\frac{3}{2}\lfloor \frac{l+1}{2} \rfloor \geqslant \frac{3}{2} * \frac{l}{2} = \frac{l}{2} + \frac{l}{4}$,因为 $l > 2$,所以 $\frac{3}{2}\lfloor \frac{l+1}{2} \rfloor \geqslant \frac{l}{2} + \frac{l}{4} \geqslant \frac{l+1}{2}$,故 $nl(k) = \frac{3}{2}\lfloor \frac{k}{2} \rfloor \geqslant$

$\frac{k}{2}$ 成立。

$\frac{1}{2}\lfloor\frac{k}{2}\rfloor < m < \lfloor\frac{k}{2}\rfloor$ 时,设最大匹配中的基因对中在步骤四正确匹配的有 u 对,不正确匹配的有 v 对,$u+v=\lfloor k/2\rfloor$。在步骤四、五总计得到的内邻接数为:

$$nl(k)=u+\frac{v}{2}*(1+2)=\frac{2(u+v)}{2}+\frac{v}{2}=\lfloor\frac{k}{2}\rfloor+\frac{v}{2}$$

因为 $v\geqslant 2$,故 $nl(k)\geqslant\frac{k}{2}$。

$m=\lfloor\frac{k}{2}\rfloor$ 时,若 k 为偶数,则在步骤四便获得 $k/2$ 个内邻接;若 k 为奇数,步骤四获得 $\lfloor k/2\rfloor$ 个内邻接后在步骤五可再获得一个内邻接,故 $\lfloor\frac{k}{2}\rfloor+1\geqslant\frac{k}{2}$。

综上所述,引理 1 成立。

引理 2　$\frac{\mathrm{OPT_I}}{\mathrm{APP_I}}\leqslant 2$

证明:步骤一中使用贪婪策略插入 1-type Ⅰ串,在最坏情况下,一个错位的 1-typeI串 s_1,会导致一个 u -type Ⅰ 串 s_u 变为 type Ⅲ,一个 v -type Ⅰ串 s_v 和一个 w -type Ⅰ串 s_w 变为 type Ⅱ,u,v,w 表示各类型串的长度。

用 OPT(s),APP(s) 分别表示基因 s 在最优解和近似解中获得的公共邻接数,则 s_1,s_u,s_v,s_w 新增公共邻接比为:

$$\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)+\mathrm{OPT}(s_v)+\mathrm{OPT}(s_w)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)+\mathrm{APP}(s_v)+\mathrm{APP}(s_w)}$$

首先看 $\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)}$ 部分:

$u=1$ 时,s_u 在最优解中可以得到两个外邻接,在本算法中变成 1-type Ⅲ串无新增公共邻接,所以

$$\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)}=\frac{2+2}{2+0}=2$$

$2\leqslant u$ 时,s_u 在步骤四、五得到 $nl(u)$ 个内邻接,所以

$$\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)}=\frac{2+u+1}{2+nl(u)}$$

$$\leqslant\frac{u+3}{2+\frac{u}{2}}\qquad\text{(根据引理 1)}$$

$$=\frac{u+3}{\frac{1}{2}(u+4)}\leqslant 2-\frac{1}{u+4}\leqslant 2$$

故 $\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)}\leqslant 2$ 恒成立。

再看 $\frac{\mathrm{OPT}(s_v)+\mathrm{OPT}(s_w)}{\mathrm{APP}(s_v)+\mathrm{APP}(s_w)}$ 部分:

$1\leqslant v,w$ 时,s_v 和 s_w 在步骤二均可获得一个外邻接。$2\leqslant v,w\leqslant 3$ 时,s_v 和 s_w 在步骤二获得外邻接后,在步骤三又均获得一个内邻接。

故 $1\leqslant v,w\leqslant 3$ 时,s_v,s_w 新增公共邻接比如下:

$$(v,w)=(1,1):\frac{2+2}{1+1}=2,$$

$$(v,w)=(1,2):\frac{2+3}{1+2}=\frac{5}{3},$$

$$(v,w)=(2,2):\frac{3+3}{2+2}=\frac{3}{2},$$

$$(v,w)=(2,3):\frac{3+4}{2+2}=\frac{7}{4},$$

$$(v,w)=(3,3):\frac{4+4}{2+2}=2$$

$4\leqslant v,w$ 时,s_v 和 s_w 在步骤三结束后都得到两个公共邻接,剩余长度为 $v-2$ 和 $w-2$ 的串在步骤四、五获得的公共邻接数分别为 nl($v-2$),nl($w-2$),所以

$$\frac{\mathrm{OPT}(s_v)+\mathrm{OPT}(s_w)}{\mathrm{APP}(s_v)+\mathrm{APP}(s_w)}=$$

$$\frac{v+1+w+1}{2+nl(v-2)+2+nl(w-2)}$$

$$\leqslant\frac{v+w+2}{\frac{v-2}{2}+\frac{w-2}{2}+4}\qquad\text{(根据引理 1)}$$

$$=\frac{v+w+2}{\frac{1}{2}(v+w+4)}=2-\frac{2}{v+w+4}\leqslant 2$$

$1\leqslant v\leqslant 3$,$4\leqslant w$ 时,$\frac{\mathrm{OPT}(s_v)}{\mathrm{APP}(s_v)}$ 比值最大为 2,故只需讨论 $\frac{\mathrm{OPT}(s_w)}{\mathrm{APP}(s_w)}$ 即可,而

$$\frac{\mathrm{OPT}(s_w)}{\mathrm{APP}(s_w)}=\frac{w+1}{2+nl(w-2)}$$

$$\leqslant\frac{w+1}{\frac{w-2}{2}+2}=\frac{w+1}{\frac{1}{2}(w+2)}\qquad\text{(根据引理 1)}$$

$$=2-\frac{1}{w+2}\leqslant 2$$

因此 $\frac{\mathrm{OPT}(s_v)+\mathrm{OPT}(s_w)}{\mathrm{APP}(s_v)+\mathrm{APP}(s_w)}\leqslant 2$ 恒成立,故 $\frac{\mathrm{OPT}(s_1)+\mathrm{OPT}(s_u)+\mathrm{OPT}(s_v)+\mathrm{OPT}(s_w)}{\mathrm{APP}(s_1)+\mathrm{APP}(s_u)+\mathrm{APP}(s_v)+\mathrm{APP}(s_w)}\leqslant 2$ 得证。

最优解中类型为 r -type Ⅰ串且没有受到错位插入影响的基因串 s_r,在步骤二后均可以获得两个外邻接,若 $r\geqslant 4$ 在步骤三后又可以获得两个内邻接。

易知 $r\leqslant 7$ 时,新增公共邻接比≤2 成立。

当 $r\geqslant 8$ 时,新增公共邻接比

$$\frac{r+1}{4+nb(r-4)}\leqslant\frac{r+1}{4+\frac{r-4}{2}}=2-\frac{2}{r+4}\leqslant 2$$

综上所述,引理 2 得证。

引理 3 $\frac{\mathrm{OPT_{Ⅱ}}}{\mathrm{APP_{Ⅱ}}} \leqslant 2$

证明:最优解中 l -type Ⅱ 串,在步骤二获得一个外邻接,若 $l \geqslant 2$ 在步骤三又可以获得一个内邻接。

易知,$1 \leqslant l \leqslant 3$ 时,新增公共邻接比 $\leqslant 2$ 显然成立。

当 $l \geqslant 4$ 时,新增公共邻接比

$$\frac{l}{2 + nb(l-2)} \leqslant \frac{l+2-2}{\frac{1}{2}(l+2)} = 2 - \frac{l}{l+2} \leqslant 2$$

综上所述,引理 3 得证。

引理 4 $\frac{\mathrm{OPT_{Ⅲ}}}{\mathrm{APP_{Ⅲ}}} \leqslant 2$

证明:由引理 1 易证得。

定理 1 $\frac{\mathrm{OPT}}{\mathrm{APP}} \leqslant 2$

证明:

$$\frac{\mathrm{OPT}}{\mathrm{APP}} = \frac{\mathrm{OPT_{Ⅰ}} + \mathrm{OPT_{Ⅱ}} + \mathrm{OPT_{Ⅲ}}}{\mathrm{APP_{Ⅰ}} + \mathrm{APP_{Ⅱ}} + \mathrm{APP_{Ⅲ}}} \tag{3}$$

$$\leqslant \frac{2\mathrm{APP_{Ⅰ}} + 2\mathrm{APP_{Ⅱ}} + 2\mathrm{APP_{Ⅲ}}}{\mathrm{APP_{Ⅰ}} + \mathrm{APP_{Ⅱ}} + \mathrm{APP_{Ⅲ}}} \text{(根据引理 2-4)}$$

$$= \frac{2(\mathrm{APP_{Ⅰ}} + \mathrm{APP_{Ⅱ}} + \mathrm{APP_{Ⅲ}})}{\mathrm{APP_{Ⅰ}} + \mathrm{APP_{Ⅱ}} + \mathrm{APP_{Ⅲ}}} = 2$$

文中算法的时间复杂度主要取决于在 X , SⅡ , D 集合元素之间求最大匹配,最坏情况下集合的 X 每个元素,要考虑每条与 SⅡ 中元素构成的无向边,接着还要考虑每条由相连的 SⅡ 中元素与 D 中元素构成的无向边,这需要 $O(|X||SII||D|)$ 时间。所以,算法时间复杂度为 $O(nml)$,其中 n,m,l 分别指缺失基因个数、两倍的 contig 数量以及断点个数。

3 算法的程序实现

完成 Contig-One-sided-SF-max 问题的多项式时间近似算法后,用 Python 语言实现了该算法,并编写了具有可视化界面的 Contig-One-sided-SF-max 问题的程序,增强人机交互体验。

程序主界面如图 7 所示。

图 7 程序主界面

用户在程序输入框中输入 G 和 S ,点击确定,程序将根据文中设计的算法自动完成基因组框架填充,并给出填充结果和新增公共邻接总数。

程序的执行如图 8 所示。

图 8 程序运行界面

4 结束语

该文重新审视了基于 contig 的单面含重复基因的基因组框架填充问题。在设计算法时以缺失基因、基因位点、断点的对应关系为依据(而不像以往研究中,只关注于其中两个之间的对应),解决了 2-近似算法的冗余公共邻接问题,又解决了 2.57-近似算法精确度较低的问题。对于近似比的优化方面,还需要进一

步研究,希望能够吸引更多人共同学习来完善这一问题。

参考文献:

[1] VASER R, ŠIKIC M. Time-and memory-efficient genome assembly with Raven[J]. Nature Computational Science, 2021,1(5):332-336.

[2] LOGSDON G A, VOLLGER M R, EICHLER E E. Long-read human genome sequencing and its applications[J]. Nature Reviews Genetics,2020,21(10):597-614.

[3] MA Jingjing, ZHU Daming, JIANG Haitao, et al. On the solution bound of two-sided scaffold filling[J]. Theoretical Computer Science,2021,2021(873):47-63.

[4] LIU Nan, ZHU Daming, JIANG Haitao, et al. A 1.5-approximation algorithm for two-sided scaffold filling[J]. Algorithmica,2016,2016(74):91-116.

[5] MUÑOZ A, ZHENG Chunfang, ZHU Qian, et al. Scaffold filling, contig fusion and comparative gene order inference[J]. BMC Bioinformatics,2010,11(1):1-15.

[6] YANCOPOULOS S, ATTIE O, FRIEDBERG R. Efficient sorting of genomic permutations by translocation, inversion and block interchange[J]. Bioinformatics, 2005, 21(16): 3340-3346.

[7] KUZMIN E, TAYLOR J S, BOONE C. Retention of duplicated genes in evolution[J]. Trends in Genetics,2022,38(1): 59-72.

[8] 李春良,宋卫星,徐勤业,等. 基于邻接的单面基因组片段填充问题研究进展[J]. 计算机应用与软件, 2021, 38(12):1-6.

[9] JIANG Haitao, ZHONG Farong, ZHU Binhai. Filling scaffolds with gene repetitions: maximizing the number of adjacencies [C]//Combinatorial pattern matching. Berlin: Springer-Verlag,2011:55-64.

[10] JIANG Haito, ZHENG Chunfang, SANKOFF D, et al. Scaffold filling under the breakpoint and related distances[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2012,9(4):1220-1229.

[11] LI Chunliang, XU Qinye, JIA Handong, et al. A polynomial time algorithm for a class of contig-based two-sided scaffold filling[C]//2020 2nd international conference on video, signal and image processing. [s.l.]:ACM,2020:85-90.

[12] 柳　楠,朱永琦,李胜华,等. 基于 Contig 的单面基因组片段填充问题研究[J]. 计算机技术与发展,2022,32(11): 8-15.

[13] JIANG Haitao, QINGGE Letu, ZHU Daming, et al. A 2-approximation algorithm for the contig-based genomic scaffold filling problem[J]. Journal of Bioinformatics and Computational Biology,2018,16(6):1850022.

[14] TAN Guanlan, FENG Qilong, MENG Xiangzhong, et al. A new approximation algorithm for contig-based genomic scaffold filling[J]. Theoretical Computer Science, 2021, 2021(853):7-15.

[15] BULTEAU L, FERTIN G, KOMUSIEWICZ C. Beyond adjacency maximization: scaffold filling for new string distances [C]//28th annual symposium on combinatorial pattern matching (CPM 2017). [s.l.]:Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik,2017.

[16] MA Jingjing, JIANG Haitao, ZHU Daming, et al. Algorithms and hardness for scaffold filling to maximize increased duo-preservations[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2021,19(4):2071-2079.