进化树结构对比问题的算法研究

Algorithm for Comparing Two Phylogenetic Trees

彭云

指导老师：肖鸣宇

2015 年 3 月 11 日

简介 - 进化树

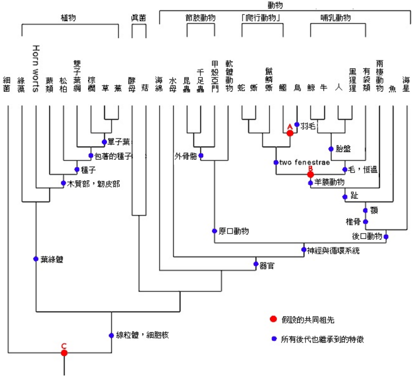
进化树 (Phylogenetic tree):

- 是表明被认为具有共同祖先的各物种间演化关系的二叉树。

简介 - 进化树

进化树 (Phylogenetic tree):

- 是表明被认为具有共同祖先的各物种间演化关系的二叉树。
- Example:



简介 - 网状事件

网状事件 (Reticulation events):

- 基因水平转移 (horizontal gene transfer)
- 重组 (recombination)
- 杂交 (hybridization)

简介 - 杂交

杂交 (Hybridzation):

- 指不同种、属或品种的动、植物进行交配

简介 - 杂交

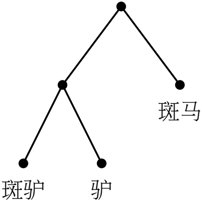
杂交 (Hybridzation):

- 指不同种、属或品种的动、植物进行交配
- Example:

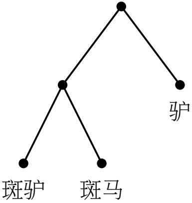
简介 - 杂交

杂交 (Hybridzation):

- 指不同种、属或品种的动、植物进行交配
- Example:



(a)



(b)

简介 - rSPR distance

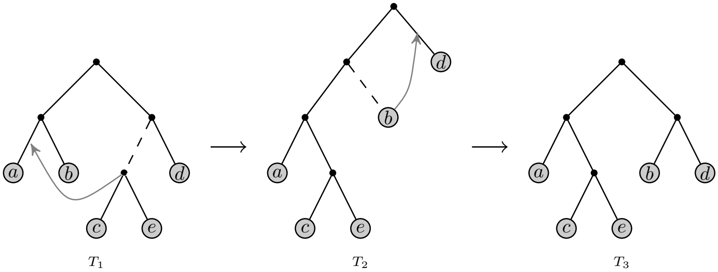
子树剪切再接距离（subtree prune and regraft distance, rSPR）：

- 将二叉树的某棵子树剪下，再接到另外一条边上，收缩掉子节点个数为 1 的所有内部节点

简介 - rSPR distance

子树剪切再接距离（subtree prune and regraft distance, rSPR）：

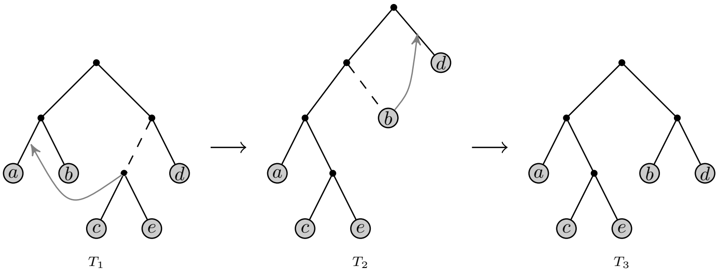
- 将二叉树的某棵子树剪下，再接到另外一条边上，收缩掉子节点个数为 1 的所有内部节点
- Example:



简介 - rSPR distance

子树剪切再接距离（subtree prune and regraft distance, rSPR）：

- 将二叉树的某棵子树剪下，再接到另外一条边上，收缩掉子节点个数为 1 的所有内部节点
- Example:

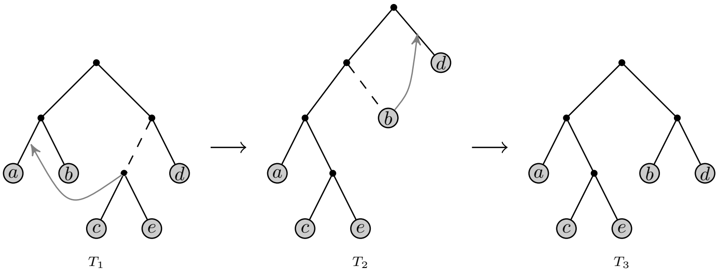


- 将其中一棵树转化为另一棵所需的最少 rSPR 操作次数称为 T_1, T_2 的 **rSPR** 距离，记作 $d(T_1, T_2)$

简介 - rSPR distance

子树剪切再接距离（subtree prune and regraft distance, rSPR）：

- 将二叉树的某棵子树剪下，再接到另外一条边上，收缩掉子节点个数为 1 的所有内部节点
- Example:



- 将其中一棵树转化为另一棵所需的最少 rSPR 操作次数称为 T_1, T_2 的 **rSPR** 距离，记作 $d(T_1, T_2)$
- 难度：NP 完全问题

方法

- 固定参数算法（Fixed Parameter Algorithm）：
  - 给出一个参数 k，通过搜索来判定两棵进化树之间的 rSPR 距离是否超过 k
  - 目前理论上最优的算法复杂度是 $O(2.344^k n)$ [1]

[1] Chen Z Z, Wang L. Faster exact algorithms for hybridization number and rSPR distance[J]. Submitted for, 2012.

[2] Whidden C, Zeh N. A unifying view on approximation and FPT of agreement forests[M]. Springer Berlin Heidelberg, 2009.

方法

- 固定参数算法（Fixed Parameter Algorithm）：
  - 给出一个参数 k，通过搜索来判定两棵进化树之间的 rSPR 距离是否超过 k
  - 目前理论上最优的算法复杂度是 $O(2.344^k n)$ [1]
- 近似算法（Approximate Algorithm）:
  - 给出一个算法，使得这个算法的结果最差不会超过最优解的 r 倍
  - 目前已经找到理论上最好的 $r = 3$ 的近似算法，复杂度为 $O(n)$ [2]

[1] Chen Z Z, Wang L. Faster exact algorithms for hybridization number and rSPR distance[J]. Submitted for, 2012.

[2] Whidden C, Zeh N. A unifying view on approximation and FPT of agreement forests[M]. Springer Berlin Heidelberg, 2009.

方法

- 固定参数算法（Fixed Parameter Algorithm）：
  - 给出一个参数 k，通过搜索来判定两棵进化树之间的 rSPR 距离是否超过 k
  - 目前理论上最优的算法复杂度是 $O(2.344^k n)$ [1]
- 近似算法（Approximate Algorithm）：
  - 给出一个算法，使得这个算法的结果最差不会超过最优解的 r 倍
  - 目前已经找到理论上最好的 $r = 3$ 的近似算法，复杂度为 $O(n)$ [2]

方向

主要研究基于固定参数算法的精确算法

[1] Chen Z Z, Wang L. Faster exact algorithms for hybridization number and rSPR distance[J]. Submitted for, 2012.

[2] Whidden C, Zeh N. A unifying view on approximation and FPT of agreement forests[M]. Springer Berlin Heidelberg, 2009.

最大一致森林 - MAF

进化树森林：

- 将进化树 T 删去若干条边 E，并对每棵子树进行收缩操作得到 F，记作 $F = T - E$，同理有 $F' = F - E$。

最大一致森林 - MAF

进化树森林：

- 将进化树 T 删去若干条边 E，并对每棵子树进行收缩操作得到 F，记作 $F = T - E$，同理有 $F' = F - E$。

最大一致森林（Maximum Agreement Forest）：

- 若存在 $E_1 \subset E_{T_1}$，$E_2 \subset E_{T_2}$，使得 $F = T_1 - E_1$ 并且 $F = T_2 - E_2$，我们称 F 为 T_1, T_2 的一致森林。

最大一致森林 - MAF

进化树森林：

- 将进化树 T 删去若干条边 E，并对每棵子树进行收缩操作得到 F，记作 $F = T - E$，同理有 $F' = F - E$。

最大一致森林（Maximum Agreement Forest）：

- 若存在 $E_1 \subset E_{T_1}$，$E_2 \subset E_{T_2}$，使得 $F = T_1 - E_1$ 并且 $F = T_2 - E_2$，我们称 F 为 T_1, T_2 的一致森林。
- 将 T_1, T_2 包含最少进化树个数的森林称为最大一致森林，其包含的进化树个数记为 $m(T_1, T_2)$

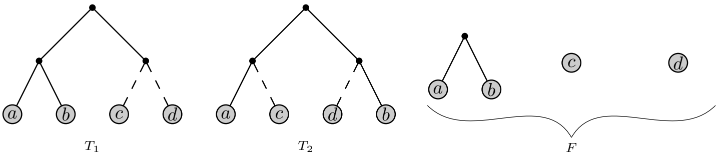
最大一致森林 - MAF

进化树森林：

- 将进化树 T 删去若干条边 E，并对每棵子树进行收缩操作得到 F，记作 $F = T - E$，同理有 $F' = F - E$。

最大一致森林（Maximum Agreement Forest）：

- 若存在 $E_1 \subset E_{T_1}$，$E_2 \subset E_{T_2}$，使得 $F = T_1 - E_1$ 并且 $F = T_2 - E_2$，我们称 F 为 T_1, T_2 的一致森林。
- 将 T_1, T_2 包含最少进化树个数的森林称为最大一致森林，其包含的进化树个数记为 $m(T_1, T_2)$
- Example:



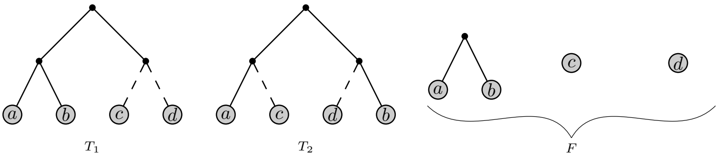
最大一致森林 - MAF

进化树森林：

- 将进化树 T 删去若干条边 E，并对每棵子树进行收缩操作得到 F，记作 $F = T - E$，同理有 $F' = F - E$。

最大一致森林（Maximum Agreement Forest）：

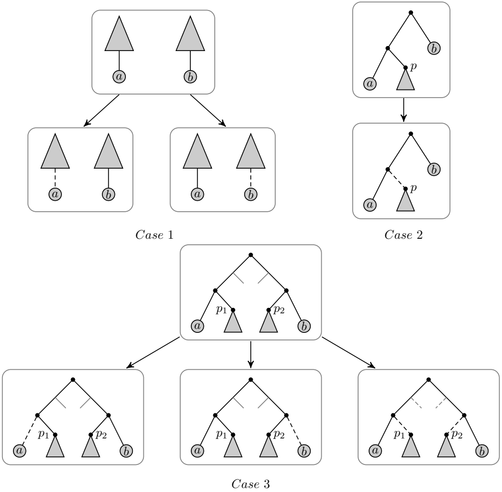
- 若存在 $E_1 \subset E_{T_1}$，$E_2 \subset E_{T_2}$，使得 $F = T_1 - E_1$ 并且 $F = T_2 - E_2$，我们称 F 为 T_1, T_2 的一致森林。
- 将 T_1, T_2 包含最少进化树个数的森林称为最大一致森林，其包含的进化树个数记为 $m(T_1, T_2)$
- Example:



- $d(T_1, T_2) = m(T_1, T_2) - 1$

Whidden 算法

在 T_1 中找一对兄弟叶节点，F_2 中分三种情况：



Whidden 算法复杂度

- 若 a, b 在 F_2 中属于不同的连通分量。

$$C(k) \leq 2C(k-1)$$

- 若 a, b 在 F_2 中连通，并且 a, b 之间有且只有一个悬挂节点。

$$C(k) \leq C(k-1)$$

- 若 a, b 在 F_2 中连通，并且 a, b 之间有至少两个悬挂节点。

$$C(k) \leq 2C(k-1) + C(k-m), \ \ m \geq 2$$

$C(k)$ 的最坏情况出现在第 3 种情况 $m = 2$ 时，此时有

$$C(k) \leq 2C(k-1) + C(k-2)$$

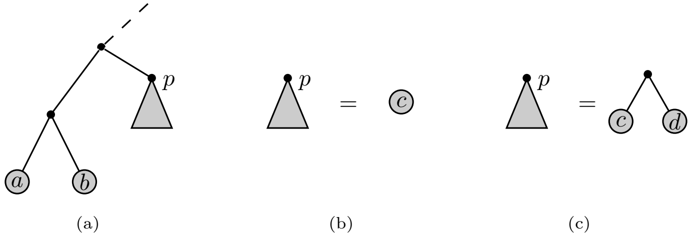
设 $C(k) = \alpha^k$，带入可得

$$1 = 2\alpha^{-1} + \alpha^{-2}$$

得 $\alpha = 1 + \sqrt{2} \approx 2.42$，复杂度为 $O(2.42^k n)$

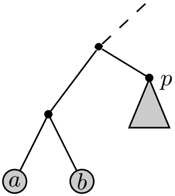
改进方法

在 T_1 中找 3 个点或 4 个点

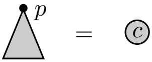


改进方法

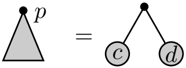
在 T_1 中找 3 个点或 4 个点



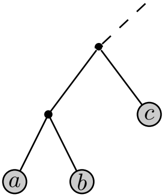
(a)



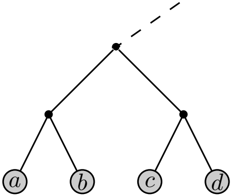
(b)



(c)



(a)



(b)

三点情况

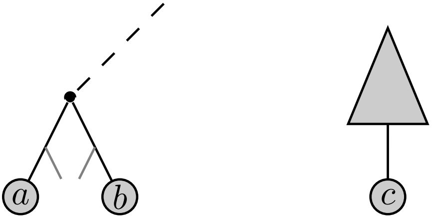
- Case 1.1：若 a, b 在 F_2 中属于不同的连通分量。

三点情况

- Case 1.1：若 a, b 在 F_2 中属于不同的连通分量。
- Case 1.2：若 a, b 在 F_2 中连通，且 $|P(a, b)| = 1$ 时。

三点情况

- Case 1.1：若 a, b 在 F_2 中属于不同的连通分量。
- Case 1.2：若 a, b 在 F_2 中连通，且 $|P(a, b)| = 1$ 时。
- Case 1.3：若 a, b 在 F_2 中连通，且 $|P(a, b)| \geq 1$ 时。



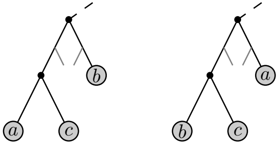
三点情况

Case 1.4：若 a, b 在 F_2 中属于相同的连通分量，满足 $|P(a, b)| \geq 2$，且 a, b 与 c 属于同一连通分量

三点情况

Case 1.4：若 a, b 在 F_2 中属于相同的连通分量，满足 $|P(a,b)| \geq 2$，且 a, b 与 c 属于同一连通分量

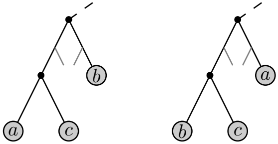
- Case 1.4.1 若 a, c 或者 b, c 是兄弟节点。



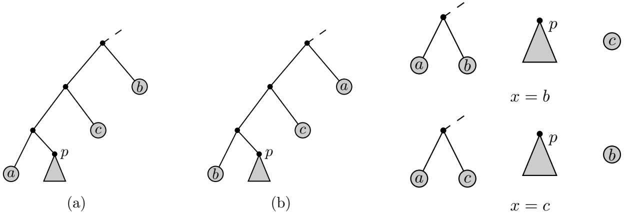
三点情况

Case 1.4：若 a, b 在 F_2 中属于相同的连通分量，满足 $|P(a,b)| \geq 2$，且 a, b 与 c 属于同一连通分量

- Case 1.4.1 若 a, c 或者 b, c 是兄弟节点。



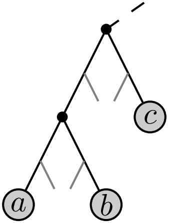
- Case 1.4.2 若 F_2 中满足 $|P(a,c)| = 1$ 且 $|P(b,c)| = 1$



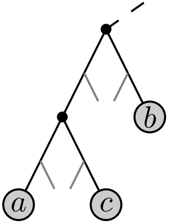
三点情况

- Case 1.4.3：排除 *Case 1.4.1* 和 *Case 1.4.2*，F_2 中一定有 $|P(a,c)| \geq 1$ 且 $|P(b,c)| \geq 1$ 且存在 $x \in \{a,b\}$ 使 $|P(x,c)| \geq 2$。不妨设 $x=a$，综合之前的条件，有

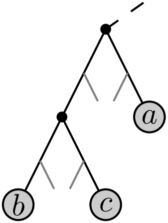
$$\begin{cases} |P(a,b)| \geq 2 \\ |P(a,c)| \geq 2 \\ |P(b,c)| \geq 1 \end{cases}$$



(a)



(b)



(c)

四点情况

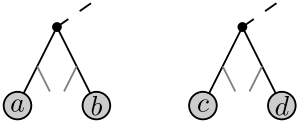
- Case 2.1：若 a, b（或 c, d）在 F_2 中属于不同的连通分量

四点情况

- Case 2.1：若 a, b（或 c, d）在 F_2 中属于不同的连通分量
- Case 2.2：若 a, b（或 c, d）在 F_2 中属于相同的连通分量，且 $|P(a, b)| = 1$（或 $|P(c, d)| = 1$）

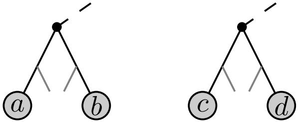
四点情况

- Case 2.1：若 a, b（或 c, d）在 F_2 中属于不同的连通分量
- Case 2.2：若 a, b（或 c, d）在 F_2 中属于相同的连通分量，且 $|P(a, b)| = 1$（或 $|P(c, d)| = 1$）
- Case 2.3：若 a, b 在 F_2 中连通，c, d 也在 F_2 中连通，且满足 $|P(a, b)| \geq 2, |P(c, d)| \geq 2$，但 a, b 与 c, d 不连通



四点情况

- Case 2.1：若 a, b（或 c, d）在 F_2 中属于不同的连通分量
- Case 2.2：若 a, b（或 c, d）在 F_2 中属于相同的连通分量，且 $|P(a, b)| = 1$（或 $|P(c, d)| = 1$）
- Case 2.3：若 a, b 在 F_2 中连通，c, d 也在 F_2 中连通，且满足 $|P(a, b)| \geq 2, |P(c, d)| \geq 2$，但 a, b 与 c, d 不连通



- Case 2.4：若 a, b 在 F_2 中连通，c, d 也在 F_2 中连通，且满足 $|P(a, b)| \geq 2, |P(c, d)| \geq 2$，且 a, b, c, d 均连通

改进算法复杂度分析

$$
\begin{cases}
C(k) \le 2C(k-1) & Cases\ 1.1, 2.1 \\
C(k) \le C(k-1) & Cases\ 1.2, 1.4.1, 2.2 \\
C(k) \le 3C(k-2) + C(k-m),\ \ m \ge 2 & Case\ 1.3 \\
C(k) \le 3C(k-2) & Case\ 1.4.2 \\
C(k) \le C(k-1) + C(k-2) + C(k-m_1) + C(k-m_2) & \\
\qquad\qquad m_1 \ge 2, m_2 \ge 3 & Case\ 1.4.3 \\
C(k) \le 2Q_1(k-1) + C(k-n_1)\ , n_1 \ge 2 & Case\ 2.3 \\
C(k) \le 2Q_2(k-1) + C(k-n_2)\ , n_2 \ge 2 & Case\ 2.4 \\
Q_1(k) \le 3C(k-2) + C(k-m),\ \ m \ge 2 & Case\ 1.3 \\
Q_2(k) \le C(k-1) + C(k-2) + C(k-m_1) + C(k-m_2) & \\
\qquad\qquad m_1 \ge 2, m_2 \ge 3 & Case\ 1.4.3
\end{cases}
$$

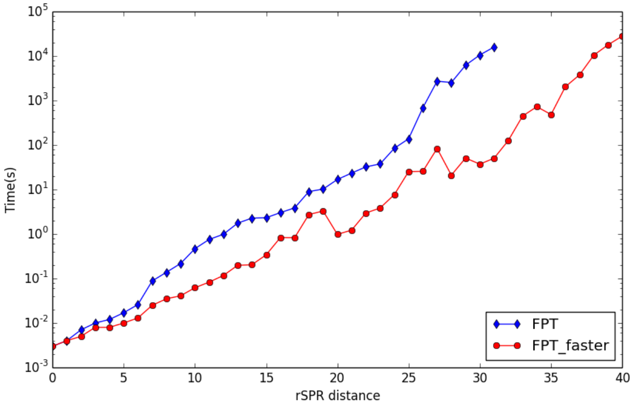
最坏情况为 Case 2.4：设 $C(k) = \alpha^k$，带入可得

$$1 = 3\alpha^{-2} + 4\alpha^{-3} + 2\alpha^{-4}$$

解得 $\alpha \approx 2.27$，改进算法的复杂度为 $O(2.27^k n)$。

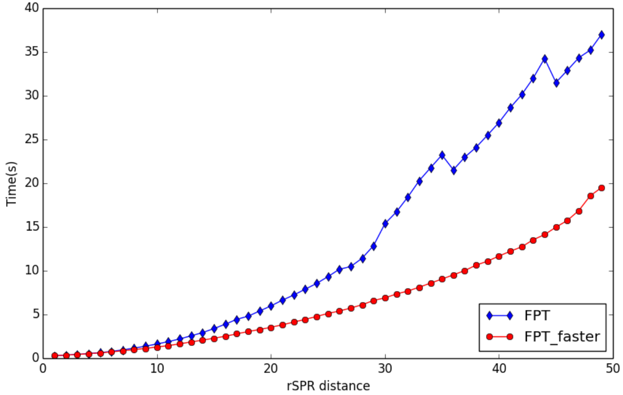
完全随机数据测试

测试了多组经完全随机 rSPR 操作后的进化树对，结果如下：



特殊随机数据测试

测试了多组经特殊 rSPR 操作后的进化树对，结果如下：



生物数据测试

真实的生物数据测试：

d_{rSPR}	组数	FPT_faster	FPT
0	8	0.001s	0.001s
1	12	0.002s	0.002s
2	4	0.002s	0.002s
3	7	0.003s	0.003s
4	9	0.003s	0.004s
5	5	0.004s	0.006s
6	2	0.005s	0.007s
7	4	0.010s	0.023s
8	2	0.021s	0.027s
9	1	0.013s	0.022s
12	2	0.173s	0.534s
17	1	3.916s	21.034s

最后，感谢各位老师在百忙之中给了我这次提前答辩的机会