工学硕士学位论文

基于距离的进化树构建算法研究

于季芝

哈尔滨工业大学

2007 年 7 月

国内图书分类号：TP301.6
U.D.C.: 681.3.06

工学硕士学位论文

基于距离的进化树构建算法研究

硕 士 研 究 生：于季芝
导　　师：郭茂祖教授
申 请 学 位：工学硕士
学 科 、 专 业：计算机科学与技术
所 在 单 位：计算机科学与技术学院
答 辩 日 期：2007 年 7 月
授予学位单位：哈尔滨工业大学

Classified Index: TP301.6
U.D.C.: 681.3.06

Dissertation for the Master Degree of Engineering

DISTANCE-BASED PHYLOGENETIC METHOD RESEARCH

Candidate: Yu Jizhi
Supervisor: Prof. Guo Maozu
Academic Degree Applied for: Master of Engineering
Speciality: Computer Science and Technology
Affiliation: School of Computer Science and Technology
Date of Defence: July, 2007
Degree-Conferring-Institution: Harbin Institute of Technology

摘　要

进化树是描述物种进化顺序和进化关系的一种拓扑结构。构建进化树在生物学中有重要的应用，研究高效准确的构建进化树算法有实际的应用价值。

本文针对进化树构建问题进行研究。进化树构建问题是一个 NP 完全问题，因而构建算法都是启发式的。通常使用的方法分为三大类：(1)距离法，(2)最大简约法，(3)最大似然法。本文使用基于最大似然法的距离法构建进化树，主要分为两个阶段：首先估计所有分类群之间的进化距离，只有序列进化距离越精确，构建的进化树才会越准确；然后基于这些距离值之间的关系构建进化树。本文工作主要包括以下两个阶段：

在第一阶段，对距离估计方法和最大似然法进行了研究，改变了传统的进化距离估计方法，把序列比对由两个序列比对扩大到多重序列比对；利用基于 quartet 的最大似然法对进化模型进行优化并使用最小二乘法快速重构所有可能的 quartet 拓扑结构，并优化其分支长度，对两两序列间距离进行估计，为进化树的构建提供了更加准确的距离矩阵。在第二阶段，主要对邻接法构建进化树进行改进，改进工作分为两部分：一是针对在第一阶段得到的距离矩阵并不总是满足可加性的问题，引入了距离的方差和协方差，采用加权的计算公式更新距离矩阵，使算法构建的拓扑结构更加准确。二是改进了邻接法的贪心特性，由于邻接法每次只聚合速率校正距离最小的两个分类单元，进而导致整个体系的偏差，实验证明速率校正距离最小的配对并不一定是在真实的进化树中进化距离最近的，基于此，改进算法每次聚合速率校正距离满足“neighbor”的两个分类单元，使算法不总是搜索分支长度之和最短的进化树，很大程度减少了这种体系偏差对真实进化关系的影响，并通过实验验证了该算法的准确性。

最后，基于以上的研究成果，实现了一个进化树构建系统。

关键词　进化树；距离法；邻接法；最大似然法

Abstract

Phylogenetic tree is a kind of typological structure for describing the sequence and relationship of species revolution. It is significant to construct phylogenetic tree in the biology field whose efficient and precise algorithms could yield to a great deal of practical value.

We study the problem of constructing Phylogenetic tree in the paper. This problem is a NP complete problem whose algorithms are all heuristic. There are usually three main methods for the problem: (1) based-distance, (2) maximum parsimony, (3) maximum likelihood. We will merge maximum likelihood method to the based-distance algorithm process of constructing phylogenetic tree on the basis of advantages of both based-distance and maximum likelihood methods. There are two main phases in the process: firstly, we estimate phylogenetic distance among all the class groups in order to make sequence phylogenetic distance more precise, so we could construct more correct phylogenetic tree.

Firstly, in the first phase of the algorithm, we study based-distance method and maximum likelihood method, change the traditional based-distance estimation method and enlarge the category of sequence alignment from double sequence to multiple sequence; we employ based-quartet maximum likelihood method to optimize phylogenetic models and use every possible topology with the maximum likelihood to estimate the distance between each two sequences so as to provide more correct distance matrix for constructing phylogenetic tree. Secondly, in the second phase, we mainly improve neighbor-joining for constructing phylogenetic tree, and there are two aspects in the process: One is distance matrix obtained on the first phase is not additve; and we introduce distance variance and covariance, renew the distance matrix by means of weighted computing formula and make the top structure built by the algorithm more correct. The other is to improve the greedy characteristics of neighbor-joining; Neighbor-joining so always aggregates two class units between which transformed distance is smallest as to lead to systematic bias. However, the experiment concludes that the couple whose transformed distance is smallest is not often the one between which the distance is not smallest in the real

phylogenetic tree. Therefore, improved method always aggregates the two units whose transformed distance conforms to “neighbor”, which does not always search the phylogenetic tree that has shortest sum of branch lengths, so the improved method heavily reduce the effect which systematic bias has on the real phylogenetic relationship. Then we testify our method’s precision by experiments.

Finally, we complete a phylogenetic tree construction system in terms of research results above.

Keywords Phylogenetic tree, based-distance, neighbor-joining, maximum likelihood

目　录

第1章　绪论

1.1　课题研究的目的与意义

自达尔文时代起，许多生物学家致力于重建地球上各生命体之间的进化历史并以进化树的形式描述这部历史。理想的途径应该是利用化石的证据，但是，由于化石保存的不完备性使得由化石记录推导出的谱系树缺乏中间环节，使大多数研究者转向比较形态学和比较生理学的方法。然而，形态和生理性状的进化如此复杂，以致不可能产生一幅进化历史的清晰图像[1]。

分子生物学的进展大大改变了这种局面。由于所有生物大多数都可用 DNA 来表示，因而人们可以通过比较 DNA 来研究他们的进化关系。近年来，在研究从病毒到人类的各种生物的进化历史中，DNA 或蛋白质序列的系统进化关系分析已经成为一个重要的工具。由于不同的基因或 DNA 片段的进化速率存在较大的差异，可以通过这些基因或 DNA 片段来研究几乎所有水平上的有机体之间的进化关系。随着进化研究的不断发展，人们逐渐建立了一套依赖于核苷酸和蛋白质序列信息的理论和方法，许多领域的科学家提出了许多的标准和方法改进他们的重构技术。

进化树是描述物种进化顺序和进化关系的一种拓扑结构。进化树的构建过程就是从生物序列的信息推断生物进化历史，重新构建出系统进化关系的过程，并把这种进化关系用进化树的形式表示出来——树的叶子节点表示各个生物序列，树枝的长度表示生物间的进化距离[2]。在过去的三十年里，生物学家们已经慢慢地把进化树的重构作为主要的研究目标。

进化树重构问题的研究是生物信息学中的一个热点，根据进化树不仅可以研究从单细胞有机体到多细胞有机体的生物进化过程，而且可以粗略估计现存的各类种属生物的进化时间。进化树反映了生命基因信息的传递的历史，因而把关于不同的有机体、基因组和分子的知识组织起来。一个重构的种系发生为解决各种问题而提供了重要信息。构建进化树在生物学中有重要的应用，研究高效准确地构建进化树算法很有实际的应用价值[3]。

在种系发生研究领域，进化树具有如下作用[4]：

1. 提供了关于世系的假设，在同化和进化约束的研究中，扮演着至关重要的角色。

2. 揭示物种形成的动力学行为和生物绝迹过程——产生和减少生物种类的两种力量。

3. 发现能够提供生物所需功能的基因簇，从而使得进化树分析能够阐述生命细胞之间的功能关系。

4. 从基因家族的序列数据库中产生功能预报。

5. 有助于疫苗、杀菌剂和抗草剂的研究与发展。

鉴于种系发生是生物学研究的如此重要部分，因此进化树构建方法研究对于重构种系发生至关重要。构建进化树的过程主要分成三个步骤：(1) 分子序列或特征数据的分析；(2) 进化树的构建；(3) 结果的检验。其中，第一步的作用是通过分析，产生距离或特征数据，为建立系统进化树提供依据；第二步主要是在第一步基础上进行进化树的构建；最后将得到的结果与标准树进行比较。本文对进化树构建算法的研究正是基于这三个步骤进行。

1.2 国内外研究现状

构建进化树是分类学的重要课题，构建的过程有助于通过物种间隐含的种系关系揭示进化动力的实质。因此进化树的重构工作的就是在序列比对的基础上，建立各个物种的进化关系，将生物合理地分成一定的类群，使得同一类群内的个体成员相似。

在构建进化树的算法中，输入的数据是 DNA 序列，这些数据可以分成两种类型：一种是距离数据(distance data)或相似性数据(similarity data)，它涉及的是成对基因、个体、群体或物种的信息；另一种特征数据(character data)，它提供了单个基因、个体、群体或物种的信息。距离数据可以通过对特征数据计算得到，但特征数据则不能由距离数据得到[5]。这些数据可以以矩阵的形式表达，它们分别用于两类不同的建树方法——距离矩阵法和基于特征的方法。

已经有很多统计学的方法可以用于分析分子数据来重构进化树。通常使用的方法分为三大类：(1) 距离法，(2) 最大简约法(maximum parsimony)，(3) 最大似然法(maximum likelihood)。其中距离矩阵法主要是距离法，基于特征的方法主要包括最大简约法和最大似然法。

在距离法或距离矩阵法中，首先获得所有分类群之间的进化距离。距离矩阵(distance matrix)是在计算得到的距离数据基础上获得的，距离的计算总体上是要依据一定的遗传模型，并能够表示出两个分类单元间的变化量。进

化树的构建质量依赖于距离估算的准确性。进化树的构建则基于这些距离值之间的关系。基于距离值的构建进化树的方法有很多，我们主要使用的是那些已经被证明能有效用于实际数据分析的方法。最主要的方法是：UPGMA法(unweighted pair-group method using an arithmetic average)、最小二乘(least squares)法、最小进化(minimum evolution)法和邻接(neighbor joining)法。

UPGMA 是最简单的距离法，使用算数平均的不加权的组对法。用这一方法构建的树有时被称为表征图(phenogram)，因为它最初在数值分类学中用于反映类群的表征相似程度。然而，当基因替代速率恒定时，UPGMA 就可用于构建进化树，并且当使用基因频率数据来重建系统发育进化树时，它比其它距离法能构建出更好的树[1]。

如果进化谱系间的核苷酸替代速率不同，UPGMA 会常常会给出错误的拓扑结构。在这种情况下，应使用一些能容许各个分支核苷酸替代速率有所不同的方法。有一类方法便是最小二乘法。其中最常用的是一般 LS 法和加权 LS 法，当变量为正态分布时，它与最大似然法同样有效。

最小进化法理论认为拓扑结构中所有分支长度估计的和为最小的拓扑结构被挑选作为最优树。Rzhetsky 和 Nei(1993)用数学方法证明了该理论，即当使用无偏的进化距离估计时，无论序列数为多少，真实的拓扑结构的所有分支长度之和将会达到最小。这是一个很好的统计特性，但是，具有最小的分支长度之和的拓扑结构并不一定是真实拓扑结构的无偏估计。

尽管最小进化法有较好的统计特性，但物种数目较大时，需要相当长的计算时间。Saitou 和 Nei 基于最小进化原理提出一种有效的构树方法，能够很快地处理大规模数据[5]。该方法称为邻接法，它通过确定距离最近(或相邻)的成对分类单元来使进化树的总长达到尽可能小。

邻接法具有很低的计算时间复杂度，因此能够处理庞大的数据集合，但是拓扑准确性不是很高。邻接法本身是一个不断聚合的过程。所以，要得到更为准确的进化树拓扑结构，人们通常提高序列间进化距离估计的准确性，估计越准确，构建的进化树才会更准确。后来，又出现了很多基于邻接方法的改进方法。但是不管使用什么样的基于距离的方法，最基本的总是距离矩阵的质量。然而传统的距离估计方法并不能总提供较为准确的进化距离估计值，并且由于贪心搜索特性，使得邻接法不能总是找到最小进化树，只能找到拓扑结构和最小进化树相似的较短的进化树[6]。

与距离法不同，基于离散特征的构建方法是首先确定一个标准，然后按这个标准去比较不同的进化树，最后选择最优的树，结果符合选择标准的最

优树可能是一个，也可能是多个。

最大简约法是一种不依赖任何进化模型的统计方法，能快速地分析出大量序列之间的系统进化关系，所构建的树中的短分支更接近真实情况[1]。它通过寻求物种间最小的变更数来完成的。用最大简约方法搜索进化树的原理是要求用最小的改变来解释所要研究的分类群之间的观察到的差异。它考察数据组中序列的多重比对结果，优化出的进化树能够利用最少的离散步骤去解释多重比对中的碱基差异。但是，这种方法却不适合处理大规模数据。

最大似然法对于假设的模型有较大依赖性[7]，并且计算量较大，但准确性高，为统计推断提供了基础，主要优点是能很好地把进化模型应用到给定的数据集合中去，比邻接法更为准确。但缺点是最大似然法的计算量非常大。然而，大量实验表明，如果使用局部优化的最大似然法来估计序列集合中的所有配对之间的距离，其时间复杂度很低，并且能够使序列之间距离估计获得较高的准确性。

基于quartet的最大似然法是一个基于局部优化的最大似然算法：把n个对象的集合S分解为大小为4的子集(每一个这样的大小为4的子集称为一个quartet)，然后利用最大似然方法来构建quartet的进化树(即使最复杂的进化树构建方法也能快速的构建出四个对象的进化树)，这种方法时间复杂度很低，并且具有很好的一致性[8]，通常用来处理规模较大的数据集合。

近来，贝叶斯推断的统计学方法也应用到这一领域[1]。贝叶斯推断法涉及两个基本概念：树的先验概率(prior probability)和后验概率(posterior probability)。树的先验概率是指对进化树未进行任何观测时的概率，具体来说就是认为所有进化树都具有相同的可能性；树的后验概率是指通过观测，进化树的条件概率，即在给定的序列数据条件下，某进化树的概率。因而后验概率最大的进化树为最优树。

此外，还有其它大量建立和搜索进化树的方法，总的来说，随着分子生物学研究的深入，越来越多序列的进化意义最终会被揭示，对各种构树方法也提出了新的挑战。

这些进化树构建算法和搜索算法将在第2章具体介绍。

1.3 本文的主要研究工作

综上分析，本文提出了一种新的算法思想，将基于 quartet 的最大似然方法应用于距离法的计算过程中，利用基于 quartet 的最大似然法来优化进

化模型并利用所有可能 quartet 组合的具有最大似然值的拓扑结构来估计每两个对象之间的进化距离，并对距离矩阵进行初始化。

对基于 quartet 的最大似然法来估计进化距离可以有效地纠正进化距离的估计。因为在最大似然法中，进化树的每个分支的边长代表核苷酸的预期替代数，它作为一个参数，是通过对给定核苷酸的最大似然函数最大化来进行估计的，即在最大似然方法中，考虑的参数不是拓扑结构而是每个拓扑结构的支长。所以能更加精确进化距离的估计[9]。

本算法对 quartet 的似然值计算进行了三点优化：首先，对整个序列集合进行序列比对，去除序列冗余，以保证最大限度的不丢失序列信息。其次，利用最小二乘法对一个 quartet 的三个拓扑结构进行近似分支长度估计，并利用该分支长度计算出似然值得近似值，以确定似然值最大的拓扑结构。最后，使用 Brent 方法优化多棵四个类树的似然值以减少计算量。

基于由 quartet 的最大似然法获得的距离矩阵并不满足可加性，利用进化距离的方差和协方差构建进化树，将对进化距离的计算进行加权处理，改进了邻接法只对可加的或接近可加的距离矩阵才能构建较为准确的拓扑结构，有效地利用了距离数据中的信息，更加准确地估计进化树的拓扑结构。

另外，为了改进邻接法的贪心特性，引入了“neighbor”的概念，将其应用在上述算法中，很大程度减少了这种贪心特性引起的体系偏差对真实的进化关系的影响，以在保持距离法快速构建进化树的基础上提高准确性。

第2章　进化树构建算法

2.1　引言

构建进化树是一个典型的 NP 完全问题。进化树的构建算法有很多种，根据所处理数据的类型，可以将系统进化树的构建方法大体上分成两大类，一类是基于距离的构建方法，其计算时间复杂度一般较低，能够处理庞大的数据集合；另一类是基于离散特征的构建方法，包括最大简约法和最大似然法，其中，最大似然法相对于最大简约法准确性更高，因此，本章将重点介绍距离法和最大似然法。

2.2　基本概念介绍

由于所有的生物的蓝图都用 DNA(某些病毒中则用 RNA)来书写，因而人们通常用 DNA 序列来研究物种的进化历史。首先，DNA 仅由 4 种碱基组成，即 A(腺嘌呤)、T(胸腺嘧啶)、G(鸟嘌呤)和 C(胞嘧啶)。进化的主要原因是基因突变，主要是由核苷酸替代、插入/缺失、重组和基因转换等引发的。核苷酸替代分为转换(transition)和颠换(transversion)。转换指的是一个嘌呤(purine，腺嘌呤或鸟嘌呤)被另一个不同的嘌呤替代，或一个嘧啶(pyrimidine，胸腺嘧啶或胞嘧啶)被另一个不同的嘧啶所替代；其他的核苷酸替代皆为颠换，如图 2-1 所示。

下面是进化树中的一些常用的专用术语：

1. 树的末端代表现代生存的物种，称为叶子节点(leaves)或外节点(external nodes)。

2. 树内的分支点叫内结(internal nodes)。

3. 两个节点之间的连接部分称为分支(branches)。

4. 达到并终止于外结的分支叫周支(peripheral branches)，未达到顶结的其他的分支叫做内支(interior branches)。

进化树构建问题描述：给定一个包含有 m 个 DNA 序列的集合 S，构建一个能够反映这 m 个 DNA 序列进化关系的树型拓扑结构。S 上的进化树特点：二叉树，且树的每个叶子节点对应于每一个序列，因此有 m 个叶子。

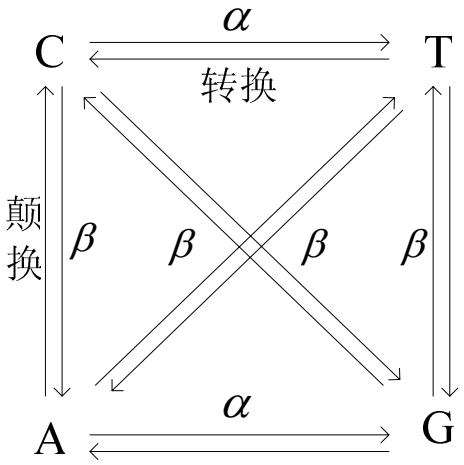

图 2-1 核苷酸的转换和颠换

Figure 2-1 Transition and transversion of nucleic acids

进化树表达的是基因或生物体的系统发育关系，所以通常也称为系统发育树，常用有根或无根的树状结构来表示。前者被称为有根树，后者被称为无根树。树的分支式样，无论有根或无根，均被称为拓扑结构。所以，进化树一般具备几个性质：(1) 如果是一棵有根树，如图 2-2 所示，则树根代表在进化历史上是最早的、并且与其它所有分类单元都有联系的分类单元；(2) 如果找不到可以作为树根的单元，则系统发育树是无根树，如图 2-3 所示。(3) 从根节点出发到任何一个节点的路径指明进化时间或者进化距离。

对于一个分类单元(即 DNA 序列)数为 m 的无根二叉树有 $2m-3$ 个分支。既然有 m 个周支连接到 m 个现存的分类单元，那么内支数即 $m-3$，内部节点数为 $m-2$。一个有根树，其内部分支数和内部节点数分别为 $m-2$ 和 $m-1$，总的分支数为 $2m-2$。对于给定的一组分类单元，有很多可能的系统进化树，但是只有一棵树是正确的，分析的目标就是要寻找这棵正确的树。

用于构建系统进化树的分子数据分成两类：(1) 距离(distances)数据，常用距离矩阵描述，表示两个数据集合之间所有的序列两两差异；(2) 特征(characters)数据，表示分子所具有的特征。离散特征数据可分为二态特征与多态特征。二态的离散特征只有两种可能的状况，即具有与不具有某种特征，通常用“0”或“1”表示。多态离散特征具有两种以上可能的状态，如核苷酸的序列信息，对序列中某一位置来说，其可能的核苷酸碱基有 A、T、G 和 C 共 4 种。可以将特征数据转换为距离数据。不过通常在将特征数

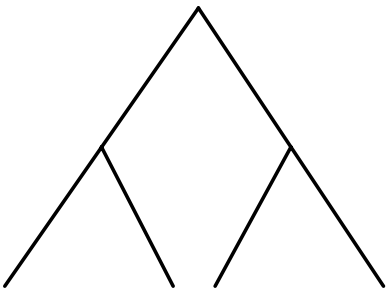

图 2-2 有根树

Figure 2-2 Rooted tree

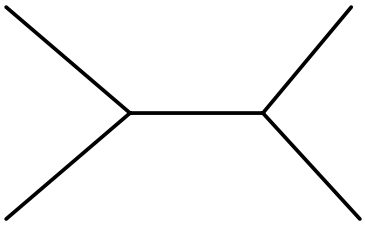

图 2-3 无根树

Figure 2-3 Unrooted tree

据转换为距离数据时，会出现信息丢失。

进化树重构工作的目的是探讨物种之间的进化关系，其分析的对象往往是一组同源的 DNA 序列。这些序列取自于不同的生物基因组的共同位点。序列比对是进行同源分析的一种基本手段，是进行系统进化分析的基础，一般采用基于两两比对的多重序列比对方法，如 Clusta1W 程序。通过序列的比对分析序列之间的距离。

无论是 DNA 序列，还是蛋白质序列，都是由特定字母表中的字符组成的，计算序列之间距离的一个前提条件是要有一个进化模型，进化模型影响距离估计的结果，影响系统进化树构建的结果。在具体分析过程中，应该选择一个合理的进化模型，参见第 3 节的各种距离打分模型。

距离是反映序列之间关系的一种度量，是建立系统进化树时所常用的一类数据。在计算距离之前，首先进行序列的预处理，去除序列冗余，进行序列比对，然后累加每个比对位置的得分。最后，应用第 3 节介绍的关于进化距离估计方法，直接计算序列之间的距离，当然也可以用其他方法产生序列之间的估计值。

2.3 基于距离的进化树构建算法

基于距离的进化树的构建算法的基本思路：确定一种序列之间的距离测度，在该距离测度下构建一棵进化树，使得该树能够很好地反映已知序列之间的距离。距离估计方法是分别对两两序列之间计算距离估计值，建立一个距离矩阵，如表 2-1 所示。然后，由基于距离的进化树构建算法根据距离矩阵构造系统进化树。

表 2-1 5 条核苷酸序列的距离矩阵

Table 2-1 Distance matrix of five pieces of nucleic acid sequences

	1	2	3	4
2	0.0516	—	—	—
3	0.0550	0.0031	—	—
4	0.0483	0.0221	0.0253	—
5	0.0582	0.0651	0.0685	0.0549

2.3.1 距离估计方法

距离估计方法产生 DNA 集合中每对序列之间的进化距离。距离法中被用来估计这些进化距离的方法有 Jukes-Cantor 方法，Kimura 的两参数法，Tajima-Nei 方法，Jin-Nei gamma distance 方法和 Tamura 方法等，其中 Jukes-Cantor 方法和 Kimura 的两参数法的应用更为广泛[10]。

2.3.1.1 Jukes-Cantor 方法 Jukes-Cantor 方法建立在一个简单的核苷酸进化模型上：由 Jukes 和 Cantor(1969)提出。该模型假定任一位点的核苷酸替代都是以相同频率发生的，且每一位点的核苷酸每年(或以任何其他时间单位)以α概率演变为其他三种核苷酸中的一种，如表 2-2 所示。

表 2-2 Jukes-Cantor 模型

Table 2-2 Jukes-Cantor model

	A	T	C	G
A	—	α	α	α
T	α	—	α	α
C	α	α	—	α
G	α	α	α	—

其中，a、β分别为两种碱基间两个不同的置换频率。因此一个核苷酸

演变为三种其他核苷酸的任何一种的概率为 $\gamma = 3\alpha$，γ 等于每年每个位点的核苷酸替换率。所以，假设 t 为两个序列 X 和 Y 的分歧时间，则两个序列每一位点的核苷酸替代期望数为 2γt。因此，序列之间的距离估计值由公式(2-1)给出。

$$d = -\frac{3}{4}\ln\left(1-\frac{4}{3}p\right) \tag{2-1}$$

式中 p——两个序列之间不同核苷酸的比例[11]。

$$p = \frac{\text{两个比对序列中相应位置不同核苷酸配对的总数}}{\max(\text{两个序列的位点长度})}$$

在此方法中，我们假设每对核苷酸的核苷酸替代率相同，所以 A、T、G 和 C 的期望频率最终会等于 0.25。然而，既然对初始频率不作假定，则公式(2-1)不需要初始频率，即不需要假定核苷酸频率不随时间变化的。

2.3.1.2 Kimura 方法 在同源假设的基础上，Jukes 和 Cantor 假设每一碱基(A、C、G、T)具有同等机率，可突变为另外三种碱基中的任何一种[12]。Kimura(1980)考虑到转换和颠换具有不同的频率(分别用 α 和 β 表示)，而且在实际数据中，转换替代速率常高于颠换速率，提出一种以 Kimura 核苷酸替代模型(如表 2-3 所示)为进化模型考虑核苷酸替代数的估计方法。

表 2-3 Kimura 模型

Table 2-3 Kimura model

	A	T	C	G
A	—	β	β	α
T	β	—	α	β
C	β	α	—	β
G	α	β	β	—

该模型中，每年每个位点转换替代速率(α)不同于颠换替代速率(2β)。应用该模型，Kimura 距离估计方法计算每两个序列之间的转换型对的总数 P 和颠换型对的总数 Q，在表 2-3 中，P 和 Q 的表达式分别如公式(2-2)、(2-3)所示。

$$P = \frac{1}{4}\left(1-2e^{-4(\alpha+\beta)t}+e^{-8\beta t}\right) \tag{2-2}$$

$$Q = \frac{1}{2}\left(1-e^{-8\beta t}\right) \tag{2-3}$$

式中 t——两个序列 X 和 Y 的分歧时间。因而，在 X 和 Y 间的每个位点核苷酸期望替代数由公式(2-4)估计。

$$d = -\frac{1}{2}\ln(1-2P-Q) - \frac{1}{4}\ln(1-2Q) \tag{2-4}$$

所以，d 的估计值可以通过观察值代替相应的 P 和 Q 而获得的。因此，在进行两两序列间的距离估计时，P 的观察值为两个序列间发生转换的位点的出现频率，Q 的观察值为两个序列之间发生颠换的位点的出现频率，如下所示：

$$P = \frac{\text{两个比对序列中发生转换的位点总数}}{\max(\text{两个序列的位点长度})}$$

$$\mathrm{Q} = \frac{\text{两个比对序列中发生颠换的位点总数}}{\max(\text{两个序列的位点长度})}$$

用 Kimura 模型，每个核苷酸的平衡频率为 0.25。然而，无论核苷酸初始频率为何，均可应用该方法类似 Jukes-Cantor 方法，不用设置核苷酸的初始频率，使得这两个模型较其他模型应用范围更广。

2.3.1.3 Tajima 和 Nei 方法 Tajima 和 Nei 在 1984 年提出了一种新的估计距离方法[13]。该方法似乎对多种干扰因子均不敏感，其基础之一是等输入(Equal-input)模型，如表 2-4 所示。

表 2-4 等输入模型

Table 2-4 Equal-input model

	A	T	C	G
A	—	αg_T	αg_C	αg_G
T	αg_A	—	αg_C	αg_G
C	αg_A	αg_T	—	αg_G
G	αg_A	αg_T	αg_C	—

其中，g_A，g_C，g_T，g_G 分别为核苷酸 A、C、T、G 在 DNA 序列中出现的频率。等输入模型是由 Felsenstein(1981)以及 Tajima 和 Nei 提出的。该方法需要假定核苷酸频率的静态分布，来估计序列之间的距离值 d。

$$d = -b\ln\left(1-\frac{p}{b}\right) \tag{2-5}$$

$$b = \frac{1}{2}\left[1-\sum_{i=1}^{4} g_i^2 + p^2/c\right] \tag{2-6}$$

$$c=\sum_{i=1}^{3}\sum_{j=i+1}^{4}\frac{X_{ij}^{2}}{2g_{i}g_{j}} \tag{2-7}$$

如果，所有的核苷酸对的替代速率都相同的话，b 在平衡时期望值为 3/4，公式(2-5)将简化为公式(2-1)。实际上，由于核苷酸替代速率不等，b 通常小于 3/4。在这种情况下，公式(2-5)将比 Jukes 和 Cantor 方法提供更大的距离值。

2.3.1.4 Tamura 方法 如前所述，Kimura 模型中的 4 种不同核苷酸最终将成为 0.25。然而，真实数据中，不同核苷酸的频率是不相等的，且 GC 含量常远离 0.5。例如，在果蝇线粒体 DNA 中，GC 含量约为 0.1[14]。考虑到这种情况，Tamura 提出了一种估计 d 值的方法[15]，其替代模型如表 2-5。

此模型是将 Kimura 二参数模型扩展到不同的 GC 含量的情况，d 的估计公式为：

$$d=-h\ln\left(1-\frac{P}{h}-Q\right)-\frac{1}{2}(1-h)\ln(1-2Q) \tag{2-8}$$

这里，$h=2\theta(1-\theta)$，θ 为 GC 含量。

表 2-5 Tamura 模型

Table 2-5 Tamura model

	A	T	C	G
A	—	$\beta\theta_2$	$\beta\theta_1$	$\alpha\theta_1$
T	$\beta\theta_2$	—	$\alpha\theta_1$	$\beta\theta_1$
C	$\beta\theta_2$	$\alpha\theta_2$	—	$\beta\theta_1$
G	$\alpha\theta_2$	$\beta\theta_2$	$\beta\theta_1$	—

2.3.1.5 Tamura 和 Nei 方法 还有一种距离估计方法，即 Tamura 和 Nei 方法。该方法运用 HKY 模型(如表 2-6)结合了 Kimura 两参数模型与等输入模型，并考虑转换/颠换和 GC 含量偏倚的情况。

然而，该模型中 d 的公式十分复杂，所以经常用 Tamura 和 Nei 模型(如表 2-6)代替 HKY 模型，在此模型中，d 的公式是：

$$d=-\frac{2g_Ag_G}{g_R}\ln\left[1-\frac{g_R}{2g_Ag_G}P_1-\frac{1}{2g_RQ}\right] \tag{2-9}$$

式中 P_1——A 对 G 的转换差的比例[16]。

表 2-6 HKY 模型

Table 2-6 HKY model

	A	T	C	G
A	—	βg_T	βg_C	αg_G
T	βg_A	—	αg_C	βg_G
C	βg_A	αg_T	—	βg_G
G	αg_A	βg_T	βg_C	—

当 $d \geq 0.6$ 时，不同的距离估计方法将获得极为不同的距离测度结果。当 d 达到 0.5 时，仅仅 Tamura 距离仍与 Tamura-Nei 距离相同，而在 $d \leq 0.25$ 时，Tamura、Kimura 以及 Jukes-Cantor 距离本质上都同 Tamura-Nei 距离相同。因此，当研究亲缘关系较近的序列时，没有必要使用复杂的距离测度，最好应用较简单的方法，因为它的方差更小。在用距离法构建进化树时，复杂的距离测度在获得正确的拓扑结构方面不一定比简单的距离测度有效。

2.3.2 距离建树方法

距离法是一种基于距离的构树方法，通过不断的迭代聚合两个聚类来决定进化树的拓扑结构和分支长度，它将进化树的构建和搜寻最优进化树的过程融合在一起，构建进化树的过程，也就是寻找最优进化树的过程[17]。

2.3.2.1 UPGMA UPGMA 是一种算术平均的不加权的组对算法。它最初在数值分类学中用于反映类群的表征相似程度。然而，当基因替代速率恒定时，UPGMA 法也可用于构建分子系统树。

为了便于分析，首先定义一个连续加和距离函数，在该函数下，两个分类单元之间的距离与系统进化树中连接这两个分类单元的分支总长度成正比。这样，如果分类单元 a 和分类单元 b 由经过中间节点 v 的两条边相连，两条边的长度分别为 d_{av} 和 d_{bv}，则它们之间的距离为 $d_{av}+d_{bv}$。

UPGMA 的基本思路是：首先从距离矩阵中选择距离最小的一对分类单元(DNA 序列)，令它们分别为 x 和 y，然后将这两个分类单元合二为一，形成一个新的分类单元(代表这两个分类单元的祖先，记为 z)，并重新计算这个新的分类单元与其它分类单元(或序列，以 u 表示)之间的距离 $d(z,u)$。其计算公式为：

$$d(z,u)=\frac{1}{2}\left(d(x,u)+d(y,u)\right) \tag{2-10}$$

每次合并所形成的新的分类单元实际上是一个聚类，以一个内部节点表示，该节点到 x 和 y 所在节点的距离相同，其值等于 $d(x,y)$ 的一半，而到其它节点的距离按照上述公式计算。每次合并后，修改距离矩阵。重复上述过程，直到所有的分类单元都被合并到一类为止。

UPGMA 的执行过程如下：

(1) 初始化：使每个分类单元(序列)自成一类，如果有 n 个分类单元，则开始时共有 n 个类，每个类的大小为 1，分别用 n 个叶节点代表每个类；

(2) 执行下列循环：

① 寻找距离矩阵中具有最小距离的两个类 x、y；

② 建立一个新的聚类 u，作为类 x 和类 y 合并后的分类单元；

③ 在树中建立一个新的内部节点 u，生成两个新的分支，将 x 和 y 连接到 u 上，使 u 作为 x 和 y 的父节点，并且将 u 到 x、y 中各个叶节点的长度赋值为 $d_{xy}/2$，计算出 d_{ux} 和 d_{uy}；

④ 按照公式(2-11)计算 u 到其它分类单元 k 之间的距离

$$d_{uk}=\frac{1}{rs}\sum_{i,j}d_{ij} \tag{2-11}$$

这里，r 和 s 分别表示分类单元 u 和 k 内的类的个数，d_{ij} 是分类单元 u 中的类 i 和分类单元 k 中的类 j 间的距离。

⑤ 更新距离矩阵：在距离矩阵中删除与 x 和 y 相应的行和列，为 u 加入新的行和列，建立新的距离矩阵；

重复循环，直到仅剩一个类为止。

UPGMA 法在不同谱系间进化速率有较大差异或有同源序列的平行进化时常得出错误的拓扑结构，而且当进化树的状态空间较大时，UPGMA 法的可操作性极差，因而该建树方法的使用极为有限。由 UPGMA 方法得到的树通常是有根树，因为在进化速率恒定的假设下很容易获得树根[18]。然而，同其它方法一样，UPGMA 是一种既构建拓扑结构又计算分支长度的方法，所以不一定要给出树根。其它系统发育推断方法构建的树通常是无根树，因为在各个分支的进化速率不相同时，很难确定树根。

2.3.2.2 最小二乘法 如果进化谱系间的核苷酸替代速率不同，UPGMA 法

通常会给出错误的拓扑结构[19]。在这种情况下，应使用一些能容许各个分支的核苷酸替代速率有所不同的方法。有一类方法便是最小二乘(LS)法。最常用的是一般 LS 法和加权 LS 法。

用于进化树拓扑结构推断的一般 LS 法中，要考虑残差平方和。设 d_{ij} 和 e_{ij} 分别是物种 i 和 j 的观察距离和先祖距离(patristic distance)。i 和 j 之间的先祖距离为在树中连接 i，j 物种的所有分支的长度估计值之和。则残差平方和公式为：

$$R_S = \sum_{i<j} \left(d_{ij} - e_{ij}\right)^2 \tag{2-12}$$

加权 LS 法是由 Fitch 和 Margoliash(1967)提出的，使用下式计算 R_S 值以挑选最终的拓扑结构。

$$R_S = \sum_{i<j} \frac{\left(d_{ij} - e_{ij}\right)^2}{d_{ij}} \tag{2-13}$$

为了计算平方残差 R_S，我们必须首先估计每个拓扑结构的分支长度和 e_{ij}。估计分支长度的简单方法是使用 Fitch 和 Margoliash(1967)方法。Fitch 和 Margoliash 方法与 UPGMA 方法建树过程是相似的，每次都选择具有最小距离的两个分类群进行聚合，直至建树完成[4]。但 Fitch 和 Margoliash 方法容许各个分支核苷酸替代速率不同，构建的进化树相对来说更加准确。虽然其估计结果和 LS 法不尽相同，但因为差别很小，所以该方法还是被普遍使用。该方法的优点是当物种数为 3 时，所有种的分支长度均可被唯一地确定。

假设物种 1、物种 2 和物种 3 的进化关系由图 2-4 给出。设 x、y、z 依次是物种 1、物种 2 和物种 3 的分支长度，则物种 1 和物种 2、物种 1 和物种 3、物种 2 和物种 3 互相之间的进化距离分别为：

$$d_{12} = x + y \tag{2-14a}$$

$$d_{13} = x + z \tag{2-14b}$$

$$d_{23} = y + z \tag{2-14c}$$

此方程组的解公式如下：

$$x = \left(d_{12} + d_{13} - d_{23}\right)/2 \tag{2-15a}$$

$$y = \left(d_{12} - d_{13} + d_{23}\right)/2 \tag{2-15b}$$

$$z = \left(-d_{12} + d_{13} + d_{23}\right)/2 \tag{2-15c}$$

这就是 LS 方法的分支长度估计过程。

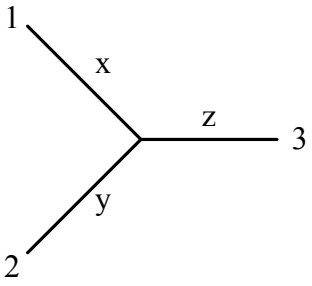

图 2-4 分支长度估计

Figure 2-4 Branch length estimation

当涉及 4 个或更多种时，首先选取两个距离最近的物种，并定义它们为 A 和 B。其余所有种合并成一个分类单元，并称为 C。物种 A 和物种 B 之间的距离不变，仍为 d_{12}，而 A 和 C 的距离现以 A 和 C 中的所有种的距离平均值表示。然后由公式(2-15a)、(2-15b)、(2-15c)得出 x、y、z。将物种 1 和物种 2 合并成复合种 AB，接着计算此复合种 AB 和所有其他物种之间的距离。从中选取除 A 和 B 以外的、距离之最小的两个种，再次被定义为 A 和 B，而包含其余所有种的复合种被称为 C。这样循环迭代，直到剩下三个种。

在 Fitch 和 Margoliash 方法中，第一个拓扑结构由上述算法得出。一旦获得这个拓扑结构，各种拓扑结构均以分支交换算法来得到，然后根据 LS 法的检验标准进行检验，最终挑选出最优树。

LS 方法是一种成熟的参数估计的统计方法[20]。当变量正态分布时，它与最大似然法同样有效。计算机模拟显示，当核苷酸数目较大时，具有非负分支限制的 LS 法能给出相当好的构树结果。

2.3.2.3 最小进化法 最小进化法(ME)的理论基础源于 Rzhetsky 和 Nei 的数学证明：当距离使用无偏估计时，树的真实拓扑结构的分支长度和的期望值最小[21]。本方法中，所有分支长度估计之和为：

$$S = \sum_{i}^{T} b_i \tag{2-16}$$

所有可能的拓扑结构都要计算出 S 值，具有最小 S 值的拓扑结构被挑选作为最优树[22]。这里，b_i 表示对第 i 支长度的估计，T 是分支的总数，即 $2m-3$，其中，m 为叶子节点数目。

如同 LS 法一样，ME 法要先检验所有可能的拓扑结构，然后找出具有最小 S 值的一个。为了达到这一目的，常用的方法是首先构建邻接树，然后对一系列与此邻接树相关的拓扑结构进行检验，以找到一个 S 值最小的树(此为暂时的 ME 树)。新的一系列与此暂时的 ME 树相近的拓扑结构(除去以被验证过的部分)再次被检验，以找到 S 值更小的进化树。此过程一直持续到没有比 S 值更小的树被发现，而具有最小 S 值的树即为最终的 ME 树。上述方法的理论基础是：当 m 值相对较小时，ME 树和 NJ 树通常很相似甚至相同，当 m 较大时，NJ 树可以作为起始树。

对分枝长度的估计除 Fitch-Margoliash 法外，还可使用最小二乘估计，Rzhetsky 和 Nei 给出了一种快速算法：分类群两两间的距离、抽样误差和分支长度分别构成列矢量 $\vec{d}$ 、$\vec{\varepsilon}$ 、$\vec{L}$，用矩阵 A 代表树的拓扑结构矩阵，行对应于分类群的两两配对，列对应于分支序数：当连接分类单元 i、j 间的路径包含第 k 个分支时，矩阵元素 $d_{(ij)k}$ 为 1，否则为 0，有：

$$\vec{d} = A\vec{L} + \vec{\varepsilon} \tag{2-17}$$

抽样误差服从平均值 0 和方差 $V(d_{ij})$ 分布，令 $T = (A^T A)^{-1} A^T$，向量 T 的第 i 行为矢量 T_i，则第 i 个分支长度 $(\widehat{L}_i)$ 的估计为：

$$\widehat{L}_i = T_i \vec{d} \tag{2-18}$$

比较所有分支长度估计的和 S，值最小的为最优树，或对任意两个树 A、B 分支长度和的差 $S_A - S_B$ 进行零假设检验：若差值大于 0，则 A 比 B 优越；若差值小于 0，则 B 比 A 优越；差值等于 0 时，则不能判断 A、B 树的好坏，对所有的树两两对比，最终获得最优树。

2.3.2.4 邻接法 尽管 ME 法有较好的统计学特性，但当物种数目较大时，需要相当长的计算时间。邻接(NJ)法是另一种快速的聚类方法，该方法是由 Saitou 和 Nei 于 1987 年基于最小进化原理首次提出的，是距离法建树中最有效的一种方法，当已知拓扑结构的树的枝长可以与进化变异的不同水平同时变化时，邻接法对预测正确的树最为可靠。

在构建进化树时，它取消了非加权分组平均法所作的基因替代速率恒定的假设，即不需要关于分子钟的假设，且在进化分支上，发生核苷酸替代的次数可以不同。邻接法选择结合后能给出分支长度最小的平方估计的序列，即能最真实的反映序列之间的真实距离，因此，邻接法是在对树的分支长度求和的基础上将序列配对。

这种方法的基本思想是：在进行类合并时，不仅要求待合并的类是相近的，同时还要求待合并的类远离其他的类。在聚类过程中，根据原始距离矩阵中所有节点间的平均趋异程度，对每两个节点间的距离进行调整，即将每个分类序列的趋异程度标准化，从而形成一个新的距离矩阵。重建时将最符合条件的两个叶节点连接起来，合并这两个叶节点所代表的分类，形成一个新的分类。在树中增加一个父节点，并在距离矩阵中加入新的分类，同时删除原来的两个分类。随后，新增加的父节点被看成为叶节点，重复上一次循环。在每一次循环过程中，都有两个叶节点被一个新的父节点所取代，两个类被合成为一个新类。整个循环直到只剩一个类为止。从得到的系统进化树来看，两个聚在一起的分类单元其所在的叶节点，到父节点的的距离并不一定相同，因为这取决于新的分支长度计算公式，不同于 UPGMA 的非加权平均的计算方法，这种方法更加准确[23]。

邻接法并不像 LS 法和 ME 法那样需要检验所有可能的拓扑结构，但在每一阶段进行物种聚合时，都要用到最小进化原理。在每一次循环中，在树中寻找最适合聚合的两个分类单元。对于节点 x，到其他节点的距离 d_x 按下式进行估算：

$$d_x = \frac{1}{n-2}\sum_{x\neq y} d_{xy} \tag{2-19}$$

式中 d_{xy}——分类单元 x 和分类单元 y 之间的距离。它是动态更新距离矩阵 D 中的元素。为了使所有的分支长度的和最小(即最小进化原理)，选择速率校正距离 $M_{xy} = d_{xy} - d_x - d_y$ 最小的一对节点 x 和 y 进行归并(如图 2-6)。

具体算法如下：

(1) 初始化：使每个分类单元自成一类，如果有 n 个分类单元，则开始时共有 n 个类，每个类大小为 1，分别用 n 个叶节点代表每个类；

(2) 执行下列循环：

① 建立一棵星状树，假设 P 为共同祖先。

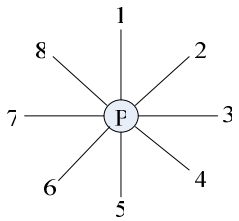

图 2-5 NJ 的星形树

Figure 2-5 A starlike tree in NJ method

② 对于所有的分类单元 x，按照公式(2-16)计算 d_x；

③ 选择一对分类单元 x 和 y，使 $d_{xy}-d_x-d_y$ 最小。

④ 将 x 和 y 归并为新的类(xy)，在树中添加一个新的节点 A，如图 2-6 所示。

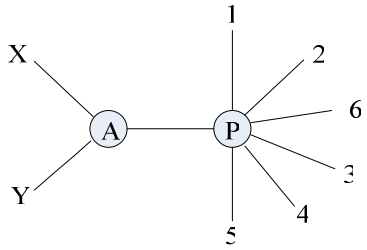

图 2-6 分类群 x 和 y 归并的树

Figure 2-6 A tree with OTUs x and y clustered

将 A 与节点 x 和 y 连接，新节点代表新生成的类，作为节点 x 和 y 的父节点，并计算从节点 x 和节点 y 到新节点(xy)的分支长度；

$$d_{x,(xy)}=\frac{1}{2}d_{xy}+\frac{1}{2}\left(d_x-d_y\right) \tag{2-20}$$

$$d_{y,(xy)}=\frac{1}{2}d_{xy}+\frac{1}{2}\left(d_y-d_x\right) \tag{2-21}$$

⑤ 计算新类与其它类 u 的距离：

$$d_{(xy),u}=\frac{1}{2}\left(d_{x,u}+d_{y,u}-d_{x,y}\right) \tag{2-22}$$

⑥ 更新距离矩阵，删除分别与 x 和 y 相关的行和列，添加新类 A。

⑦ 如果有两个以上的分类存在，则继续执行循环；否则，直接连接剩余的两个类。

NJ 法是所有基于距离建树算法中最可靠的一种方法，是所有构建发生树算法中最高效的方法，因其能快速处理大规模数据使邻接法得到了广泛的应用。但是，邻接法还存在一些缺点，需要进一步的完善。

2.4 基于特征的进化树构建算法

基于特征的进化树构建的一般问题是：给定 *n* 个分类单元，*m* 个用以描述分类单元的特征，以及每个分类单元所对应的特征值，构建一棵进化树，使得某个目标函数最大。

在构建进化树的过程中，假设特征是相互独立的，即一个特征的变化不影响另一个特征；并且在进化过程中，两个物种分叉后独立进化，互不影响。基于特征的进化树构建算法的输入一般为一个 $n \times m$ 的矩阵 M，其中 M_{ij} 代表第 i 个分类单元的第 j 个特征的取值，如表 2-7 所示：

表 2-7 特征矩阵

Table 2-7 Character matrix

物种	位点 1	位点 2	位点 3	位点 4
1	A	C	T	G
2	A	G	T	G
3	A	C	T	A
4	T	C	T	G

基于特征的进化树构建算法主要包括最大简约法和最大似然法，将在下面的章节中介绍。

2.4.1 最大简约法

最大简约法的理论基础是奥卡姆(Ockham)哲学原则，即解释一个过程的最好的理论是所需假设数目最少的那一个。如果对系统进化推断所需要知道的进化过程愈少，结果就愈可信。最大简约法的目标是构造一棵反映分类单元之间最小变化的系统进化树。

最大简约法考虑 4 个或者 4 个以上的核苷酸序列。假设 4 种核苷酸可以突变为与自身不同的任何一种，这样对于任一给定的拓扑结构，可以推断出每个位点的祖先状态。对这一拓扑结构，可以计算出用来解释整个进化过程中所需的核苷酸的最小替代数目。对于所有可能正确的拓扑结构进行这种计算并挑选出所需替代数最小的拓扑结构作为最优进化树。

2.4.1.1 最小替代数目的估计 利用最大简约方法构建系统进化树，实际上是一个对给定分类单元所有可能的树进行比较的过程，针对某一个可能的树，首先对每个位点祖先序列的核苷酸组成做出推断，然后统计每个位点用来阐明差异的核苷酸最小替换数目。在整个树中，所有简约信息位点最小核苷酸替换数的总和称为树的长度或树的代价。通过比较所有可能树，选择其中长度最小、代价最小的树作为最终的系统进化树，即最大简约树。所有的简约法程序在开始时都将会将这样一条规则应用于输入数据集：如果一个位点是信息位点，那么它至少有两种不同的核苷酸，并且这些核苷酸至少出现两次。所谓简约就是使沿着各个分支累加特征变化的数目的计算代价最小。

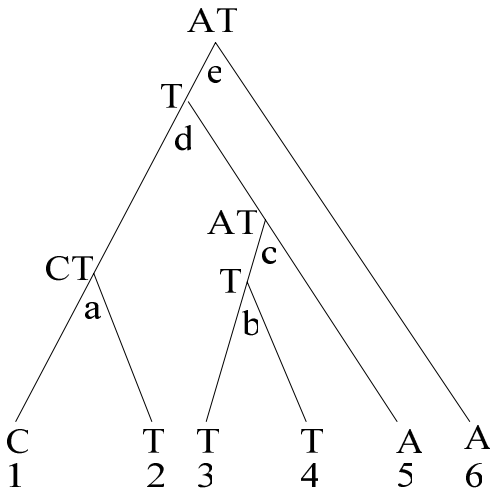

图 2-7 最小替代数目估计

Figure 2-7 The minimum substitution number estimation

如何计算或者估计一个特定的拓扑结构的最小替代数。图 2-7 中给出含有 6 个 DNA 序列的有根树的拓扑结构。这 6 个序列的某个特定位点的核苷酸的状态在系统树的外在节点上显现，其中有 1 个 C，3 个 T 和 2 个 A。从这些核苷酸中，可以知道它们起源于 5 个祖先分类节点 a、b、c、d 和 e。如果考虑最小的可能替代数，则节点 a 的核苷酸一定是 C 或 T。节点 b 的核苷酸为 T，节点 c 的核苷酸一定是 A 或 T。而节点 d 为 T，因为它的上一级祖先节点(a 和 c)都有 T。最后得到节点 e 的核苷酸为 A 或 T。通过假设所有祖先分类节点均有核苷酸 T，可以得到这些分类群的最小替代数，这个数为 3。然而，这个祖先节点核苷酸并非是对此核苷酸进化最小替代数的唯一可能的解释。

从编程的角度计算祖先核苷酸位置的算法如下：如果一个内部节点的两个直接后代节点上的核苷酸的交集非空，那么这个节点的最可能的候选核苷酸集就是这个交集；否则为它的两个后代节点上核苷酸的并集。当一个并集成为一个节点的核苷酸集时，通向该节点的分支的某个位置上必定发生一个核苷酸替换。因此，并集中核苷酸的数目也是生成外部节点上的核苷酸的最小替代数，外部节点从它们的共同祖先出发，通过这些替换，形成当前的核苷酸状态。如果需要计算一棵树在非信息位点的最小替代数，只需要把外部节点上不同核苷酸的数目减去 1 就可以了。

上例中仅仅考虑了一种拓扑结构，但在实际应用中，必须考虑所有可能的正确的拓扑结构，从而找出所需替代数最小的结构。如果对于所有的位点

和所有的结构都进行计算，可以算出每个拓扑结构的所有位点的最小替代数。这个数字我们成为树长。最大简约(MP)树即具有最短树长的拓扑结构[24]。实际上，很可能存在两种或两种以上的不同结构具有相同的最小替代数。在这种情况下，不能唯一地确定最终的拓扑结构，而所有等价简约的MP 树都可能是正确的。

2.4.1.2 最大简约树的搜索策略 穷尽式搜索是最为简单也最为直观的一种 MP 树搜索策略。当所分析的序列数或类群数(m)较小时，就可以计算出所有可能的树的长度并确定 MP 树[25]。然而，可能的拓扑结构数是随着 m 的增长而快速增长的，如公式(2-23)所示。

$$1\times 3\times 5\cdots(2m-3)=\frac{(2m-3)!}{2^{m-2}(m-2)!} \tag{2-23}$$

公式(2-23)表示的是一个类群数目为 m 的有根二叉树的可能的拓扑结构数。因而，当 m 很大时，就基本不可能检查所有的拓扑结构。但是如果我们清楚的知道若干不正确的拓扑结构，就不必计算它们的树长。只需计算可能正确的拓扑结构的树长。所以就出现了启发式的搜索方法。

分支限界法就是一种有效的启发式的搜索方法，它提供了一个逻辑方法，以确定哪些进化树值得评估，哪些可以简单地丢弃。因此，分支限界法比穷举法要快得多[25]。分支限界法最早由 Hardy 和 Penny 于 1982 年提出。

分支限界法包含两个基本步骤：第一步是为最大简约树的长度设定一个上限 L。L 的值可以采用随机选择的任何一棵初始进化树的长度。第二步是进化树的生长过程，即在描述部分物种之间关系的树中每次增加一个分支。这个方法的原理是，由数据子集得到的任何一棵树，如果它的替换数大于 L，那么当剩下的序列加入后，总的分支长度必定会变得更大。在分析过程中，如果发现比建立初始上限的树替换数更小的树，L 的值将随之修正。和穷举搜索一样，分支限界法保证在分析完成时没有遗漏更简约的树，它还有比穷举搜索方法快几个数量级的优点，所以能够用于分析多达 20 条序列。但是对于更多可能的无根树的分析，这还是不够的。

还有一种常使用的启发式方法是采用逐步加入的方式来寻找最大简约树。首先建立 3 个物种的初始树，接下来将第四个物种插入到初始树的 3 个分支中的一个，用 MP 法计算树长。对另外两个分支也进行同样的计算，记录 3 个树长中的最小值。然后，有最小的树长的 4 个物种的树将用于下一个物种插入。下一个物种再连接此 4 个物种的当前树的每个分支，然后具有最小的树长的 5 个物种的树被选中。重复这一过程，直至包含所有物种的进化

树产生为止，这个是临时的 MP 树。接下来，采用分支交换策略来找到一个树长更小的树，满足一定条件则停止交换。此时得到的进化树就是最大简约树。

启发式算法还包括模拟退火等算法，这类算法不一定总能找到最大简约树，得到的往往是一个局部最优结果，但接近于全局最优。大量序列比对的可能的无根树的数目是一个天文数字，甚至连研究其中的一小部分都是很困难的，启发式搜索通过交换分支产生更短的树，从而解决了这个问题。许多搜索最大简约树的启发式方法都基于同样的假设：最大简约树应该与次简约树有相似的拓扑结构。这些算法都是首先构造一棵初始进化树，从它开始寻找树长更短的树。如果初始的进化树很接近于最大简约树，启发式搜索会更有效。

2.4.2 最大似然法

最大似然法(ML)是一种比较成熟的参数估计的统计学方法。最大似然法基于两条基本假设：不同的性状进化是独立的；物种发生分歧后进化独立。单个位点的似然值是指在核苷酸替代模型中该位点每个可能被取代或再现的概率之和，将所有位点似然值相乘就得到进化树的似然值。当样本很大时，似然法将获得参数估计的最小方差。如果可以用公式表达一个涉及未知参数的似然函数时，就可以用最大似然法求解未知参数。

最大似然法是以一个特定的模型分析性状矩阵，使所获得的每一个拓扑结构的似然率最大，挑出最大似然率最大的拓扑结构作为最终树[26]。所考虑的参数不是拓扑结构，而是每个拓扑结构中的分支长度，并对似然函数求最大值来估计它。与距离法和简约法相比，似然法更准确一些。

利用最大似然法来推断一组序列的进化树，需要首先确定序列进化的模型，然后基于一定的模型考虑物种序列之间的关系，计算分支长度。这个过程需要寻找在某一进化距离上由第一种序列真正转化成第二种序列的可能性，并确定在最大可能下的进化距离。接着生成多个物种所构成的所有可能树，对每个树的统计量进行似然估计。最后通过树长度的优化，从而获得最优树各参数的最大似然估计。

2.4.2.1 似然值的计算 对于一棵给定的树，应该有一种评价方法，可以评价所得到的进化树 T。对于一组给定的分类单元，假设他们的观察值为 M，可以选择一棵树，使得 $P(M|T)$ 最大，即最大似然法。

现在已经知道了进化树 T 的拓扑结构，目标是计算进化树 T 的似然值，寻找树 T 的最优分支长度。考虑一个简单的四类群的树(如图 2-9)，并假设这个 DNA 序列为 n 个核苷酸位点长度，且没有插入或缺失。已知序列 1、2、3、4 的某个位点(设为第 k 个位点)的核苷酸分别为 x_1、x_2、x_3和 x_4，但并不知道节点 5 和 6 的核苷酸，而是假设他们分别为 x_5和 x_6。这里，x_i代表 A、C、G、T 中的任何一种核苷酸。

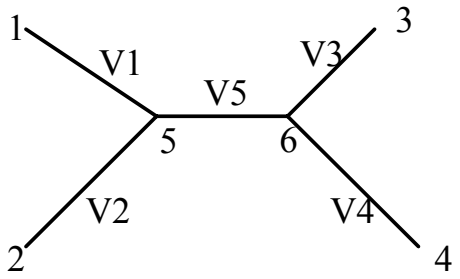

图 2-8 四个分类群的无根树

Figure 2-8 An unrooted ML tree with four OTUs

考虑一个核苷酸位点，令 $P_{ij}(t)$表示给定位点在时间 0 时的核苷酸 i 到时间 t 时的核苷酸 j 的概率。这里，i、j 指的是 A、C、G、T 中任一种。在 ML 法中，允许各分支的替代的速率(r)不同，因而根据预期的替代数($V=rt$)来估计进化时间就很方便。用$V_i \equiv r_i t_i$来表示第 i 个分支上的预期替代数。在 ML 法中，分支长度V_i作为一个参数，通过对给定核苷酸的似然函数最大化来进行估计。这样，一个核苷酸位点(第 k 个位点)的似然函数为：

$$l_k = g_{x_5} P_{x_5 x_1}(v_1) P_{X_5 X 2}(v_2) P_{X_5 X_6}(v_5) P_{X_6 X_3}(v_3) P_{X_6 X_4}(v_4) \tag{2-24}$$

式中 g_{x_i}——节点 i 为核苷酸x_i时的先验概率。g_{x_i}常常等于核苷酸x_i在整个序列中的相对频率，它可以用 ML 法来估计。

在进行公式计算时，通常用一个可逆的核苷酸替代模型来定义$P_{ij}(t)$来构建无根进化树。可逆模型就是说从时间 0 到时间 t 之间，核苷酸替代的过程是可逆的，数学上，这个可逆条件为：

$$g_i P_{ij}(v) = g_j P_{ji}(v) \tag{2-25}$$

对任意的 i 和 j，公式(2-25)要满足这个条件，并且这个似然率为节点 5 和 6 的所有可能的核苷酸之总和。

$$l_k = \sum_{x_5}\sum_{x_6} g_{X_5} P_{X_5X_1}(v_1) P_{X_5X2}(v_2) P_{X_5X_6}(v_5) P_{X_6X_3}(v_3) P_{X_6X_4}(v_4) \tag{2-26}$$

然而，上式只是第 k 个位点的核苷酸的似然率，实际上所需要的是包括不变位点在内的所有核苷酸位点。整个序列的似然值 L 是对所有位点的 L_k 求积，则整个树的似然率通常用其对数形式来表示：

$$\ln L = \sum_{k=1}^{n} \ln L_K \tag{2-27}$$

接下来，通过改变参数 v_i 使 $\ln L$ 的值最大化。有多种方法可以求解 $\ln L$ 的最大化，常用的有 Newton 法、梯度下降法和 EM 算法。ML 树是具有最大 ML 值的拓扑结构，这个拓扑结构的分支长度由提供最大似然率的 ML 估计的参数 v_i 确定。

2.4.2.2 最大似然树的搜索策略 最大似然法考察输入数据中序列的多重比对结果，优化出拥有一定拓扑结构和枝长的进化树，这个进化树能够以最大的概率导致考察的多重比对结果。因此最大似然法的搜索策略与最大简约法是相似的。

当物种数目小于 10 时，可以采用穷举法来寻找最大似然树。但由于单一的进化树的数量会随着分类物种数量的增长而呈指数增长，因此这种方法也只适用于物种数目很小的情况。当物种数目较多时，比如分析 10 个物种，至少要考虑 200 万棵树；超过 12 个物种，即使是最快的计算机也不能完成这个穷举搜索。和简约法相似，当物种数目比较大的时候，必须采用近似而更为有效的算法，常用的是启发式方法。启发式方法包括重复实验方法和分支交换方法等。它们都是基于这样的假设：各个可能的树并不总是相互独立的。因为最大似然树应该和次优的似然树有相似的拓扑结构[26]。基于似然法的启发式搜索并不逐个分支地构建所有可能的树，而是通常采用逐步加入的方式来建立初始进化树，基于一定的搜索策略寻找最大似然树。从一棵只有三个物种的树开始，一个一个逐步的加入新的物种。在每一步中选出一个新的物种，考虑其在当前的树中所有可能的位置，选出那个最有可能的位置，然后进行下一步。直到所有的物种都已经处理完。最后得到的树就是最大似然树[37]。

2.5 构建进化树方法的比较

当序列间的分歧度不高且序列较多时，距离法、最大简约法、最大似然

法所构建的系统树往往具有相似的拓扑结构，然而，一般应用这些方法所构建的进化树常有拓扑差异，因此有必要比较几种构建进化树方法，从而对特定的序列选择合适的构建进化树方法。

UPGMA 法在不同谱系间进化速率有较大差异或有同源序列的平行进化时常得出错误的拓扑结构，而且当进化树的状态空间较大时，UPGMA 法的可操作性极差，因而该建树方法的使用极为有限。

NJ 法的运算速度最快，但该算法每迭代运算一次均只搜索最近邻居配对，对其他可能的配对不加考虑，最终只生成单一的最优树，可能会遗漏一些拓扑结构更合理的次优树为弥补缺陷，William 等提出遍及邻接法(Generalized NJ)，将一些最接近的树也包含在搜索算法中[27]，逐步迭加，期望找到更好的 NJ 树。

ME 与 FM 法从所有可能的进化树中挑选树长最短的作为最优树，是距离法建树中两种相对较好的方法。不过，值得注意的是距离法在将原始数据转换成距离矩阵时难免会丢失一些进化信息。

最大简约法是一种不依赖任何进化模型的无噪声统计方法[28]，能快速地分析出大量序列之间的系统发育关系，所构建的树中的短分支更接近真实。但简约树的分值完全决定于所有重建祖先序列中的最小突变数，而突变是否按照事先约定的核苷酸最少替代的途径进行是不得而知的，单一的突变图谱可能会得出似是而非的结论[29]。因此，当序列单位位点上核苷酸替代数相对较大时，MP 法则极可能得出错误拓扑结构的树。

最大似然法似乎是几种常用方法中最为近似的方法，它考虑了所有可能的突变路径，能完全利用数据的系统发育信息。似然法运算强度极大，对于分类群较多时十分费时，这也是似然法应用的最大障碍，而且似然法并没有评估拓扑结构的优劣，分支长度估计最精确的拓扑结构为最优树。

2.6 本章小结

本章主要介绍了构建进化树的算法，根据处理数据的不同，把算法分成两大类：一类是基于距离的进化树构建算法，一类是基于特征的进化树构建算法。所以本章首先简要介绍了一些要涉及到的基础知识，然后介绍了基于距离的几种方法，接着简要介绍了基于特征的最大简约法和最大似然法，并将这些算法作了比较。

第3章　基于距离的进化树构建改进算法

3.1　引言

进化树构建主要分三个步骤：(1) DNA 序列或特征数据的分析；(2) 系统进化树的构建过程；(3) 结果的检验。其中，第一步是通过分析，产生距离或特征数据，为建立进化树提供依据。在第二步中，基于距离的进化树构建算法，因其能够快速处理大批量数据而被人们广泛使用，在基于距离的进化树的重构方法中，邻接算法是最为流行的算法之一，被认为是一种高效算法，能够快速处理大批量数据[30]，然而，在构建进化树的准确性却存在不足，本章主要介绍对邻接法的改进：在第一步中，使用基于 quartet 的最大似然法提供对距离矩阵初始化；在第二步中，引入进化距离的方差和协方差以及“neighbor”改进邻接法的构树过程；最后，对改进算法进行实验，并分析结果。

3.2　距离估计方法

3.2.1　距离估计方法分析

距离法首先需要估计序列之间的进化距离，初始化距离矩阵，然后基于距离构建进化树。因此，最基本的总是距离矩阵的估计质量，只有进化距离估计准确，才有可能构建出更加准确的进化树的拓扑结构。

距离估计方法产生 DNA 集合中每对序列之间的进化距离。距离法中被用来估计这些进化距离的方法有 Jukes-Cantor 方法，Kimura 的两参数法，Tajima-Nei 方法，Jin-Nei gamma distance 方法，和 Tamura 方法等，其中 Jukes-Cantor 方法和 Kimura 的两参数法的应用更为广泛。

这些进化距离估计方法是通过累加各种配对，把这些配对的数目代入特定的进化距离估计公式，以完成对进化距离的估计。为了便于分析，我们看一个例子。

对于序列 T1 和 T2：

T1：　AACGTCCATCGGTGA

T2： AACCATCATCCTGGA

以 Jukes-Cantor 方法为例，如前面所述，进化距离的估计过程是：估计物种 T1 和物种 T2 之间的距离时，首先对序列进行预处理，只考虑序列之间的信息位点(能够把两个序列区分开来的位点)，舍弃非信息位点。所以两个序列之间的信息位点数目是 6，然后将其代入距离估计计算公式。

当序列很相近时，他们之间的观察距离和实际距离是相当的，即发生在位点上的核苷酸的可能的替换次数较少时，由上述方法得到的距离估计值可以直接根据信息位点的个数来计算，可以保证获得距离估计值能够较为真实的反映实际进化距离。

随着时间的推移，序列之间的分歧也越来越多，发生在一个位点上的核苷酸的可能的替换次数也随之增多，出现了单个位点上的平行突变或回复突变等多重替代，当某个位点上发生与原来序列相同的核苷酸替代时，导致这些方法对现存物种的序列间的可观察到的信息位点减少，忽略了非信息位点中隐含的序列信息，不能正确地反映物种之间的进化距离。所以在上面提到的进化距离的估计方法中，每次只对两个序列进行观察，并且利用的只是存在于核苷酸的序列差异的信息位点数，对于所有序列，没有考虑单个位点上的多重替代，所以它们的距离估计是不准确的，通常会低估了真实的进化距离。

这就是传统的进化距离估计的缺点，即在将 DNA 序列转化为表示距离的数值时，在上述的进化距离估计方法下，由于在距离估计时观察范围的局限性以及核苷酸的多重替代，遗漏了许多隐藏着核苷酸替代事件的表现为非信息位点的位点，因此将很多序列信息丢失，造成序列之间距离估计的不准确。

在最大似然方法中，把序列比对由两个序列比对扩大到多重序列比对，这样可观察到的序列信息大大增多[31]。而且在似然函数中，考虑的不仅是两个物种之间的进化关系，而是由两个或两个以上序列组成的进化树的拓扑结构，其似然值的计算是基于拓扑结构的所有位点的似然值的乘积，对于每个位点的似然值，还计算了每个内部节点的所有可能的核苷酸替代，这样就可以考虑更多位点的核苷酸的平行突变或恢复突变等多重替代情况；在计算似然率时，所得到的似然函数的每个参数为核苷酸的替代数，似然值的最大化是通过数值计算实现的，所以对核苷酸替代数的估计过程即序列之间距离值的估计过程转化为似然函数的最大化过程，并不仅仅依赖于对序列的观察值。

利用最大似然能够很大程度地提高进化距离估计的准确性，然而当使用一个较大的序列集合时，计算所有可能的拓扑结构的似然值计算量是非常大的，不可能在一个合理的时间内解决[32]。有许多方法致力于提高最大似然法的计算速度。但是不管对原始的最大似然方法如何进行改进以加快其速度，最大似然法仍然是很慢，以致于只可用来处理较小的数据集合[33]。

大量仿真实验证明，使用局部优化的最大似然法来估计序列集合中的所有配对之间的距离时间复杂度很低，并且能够使序列之间距离估计获得较高的准确性。在此基础上，在包含有更多的序列的情况下来估计两个对象的进化距离，得到的进化距离会更加准确[34]。因为，对较小的数据子集使用最大似然方法分析，允许运用更加复杂的进化模型，又因为最大似然方法的主要优点能够将给定的数据集合很好的应用于进化模型中，进而能提供更加精确的分析，所以，在后面的改进的距离估计方法采用基于局部优化的最大似然法，对于用来构建进化树的距离法来说，距离矩阵的质量提高了，它能够很好地根据所获得距离矩阵构建更加准确的进化树[9]。

3.2.2 基于 QUARTET 的距离估计

虽然最大似然方法处理一般的 n 个序列的数据算法复杂度很高，但是使用局部优化的最大似然法来估计序列之间距离能够获得很高的准确性，且对于确定少量序列的最大似然树不用考虑似然值计算的时间复杂度的[7]。

改进算法中，引入基于 quartet 的最大似然法来估计进化距离，有效地纠正进化距离的估计。首先，quartet 的定义是：一个 quartet 是一棵基于四个类上的无根二叉树。对于每一个 quartet，都能够推断出四个类的唯一的无根的二叉拓扑结构。

对于每一个 quartet 组合，设它包含的四个类是$\{A,B,C,D\}\subseteq S$，我们使用$\{AB|CD\}$表示具有配对(A，B)和配对(C，D)的 quartet(如图 3-1)。一个 quartet$\{AB|CD\}$是和这样一棵进化树 T 一致的：quartet 具有的四个类在进化树 T 中都是叶子节点，并且树 T 中从 A 到 B 的路径并不和从 C 到 D 的路径交叉。同样的，如果进化树 T 的一棵包含四个类的同源子树本身就是一个 quartet，那么 quartet$\{AB|CD\}$就和进化树 T 是一致的。如果 quartet$\{AB|CD\}$不和进化树 T 是一致的，则$\{AB|CD\}$对于进化树 T 是一个错误的拓扑结构，图 3-2 就阐明了这个思想。让$Q(T)=\{\{AB|CD\}|A,B,C,D\in S且\{AB|CD\}和T是一致的\}$。如果$Q(T)$表示所

有和进化树 T 一致的 quartet 的集合，则$Q(T)$能唯一表示进化树 T，并且能在多项式时间内由进化树 T 重构[35]。要重构基于不同的四个类集合的所有 quartet 集合的 Q，流行的方法就是使用在计算上具有较高敏感性和在统计上具有一致性的最大似然法[36]。

对于每个 quartet 组合，并不是每个拓扑结构都是准确的，因此有必要使用最大似然方法去掉不正确的拓扑结构[8]。每个 quartet 组合都有三种可能的拓扑结构Q_1，Q_2，Q_3，如图 3-1 所示，每一个相应的似然值m_1，m_2，m_3，然后选择这个三个拓扑结构中具有最大似然值$m_i = \max(m_1, m_2, m_3)$的一个$Q_i$作为该 quartet 的进化树的拓扑结构。

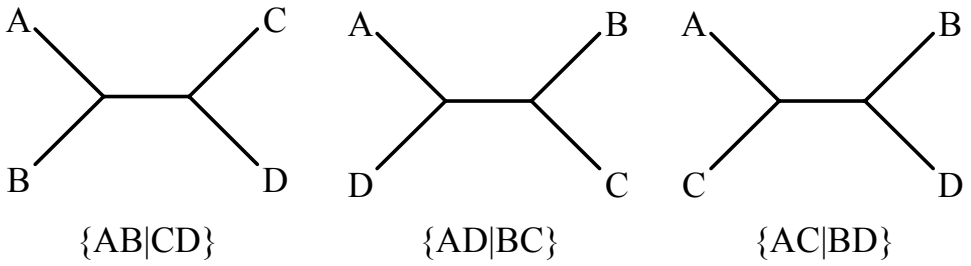

图 3-1 四类树的三种可能的拓扑结构

Figure 3-1 The three possible topologies for a four-taxa tree

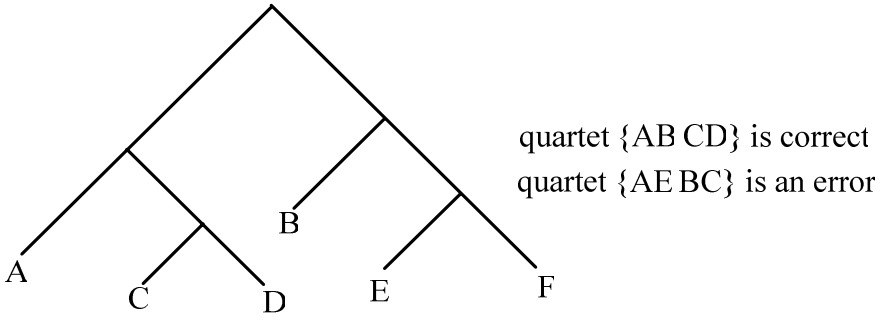

图 3-2 {AB|CD}是和该树一致的 {AE|BC}关于树是错误的

Figure 3-2 The quartet {AB|CD} agrees with the tree
while the quartet {AE|BC} is an error with respect to the tree

所以，基于 quartet 的最大似然方法很容易确定每个 quartet 的最大似然树。在计算这个最大似然树的过程中，进化树的每个分支的边长代表核苷酸的预期替代数，它作为一个参数，是通过对给定核苷酸的最大似然函数最大化来进行估计的，即在最大似然方法中，考虑的参数不是拓扑结构而是每个拓扑结构的支长。所以能更加精确进化距离的估计。在给定进化模型下，最

大似然方法能使每个 quartet 保持统计上的一致性、健壮性[8]。

新的进化距离的估计的具体方法是，利用最大似然法重构每个可能的 quartet 组合的具有最大似然值的拓扑结构，基于这个拓扑结构精确计算每个分支长度，累加拓扑结构中每两个叶子节点之间的分支长度到距离矩阵中对应位置，以此精确两物种之间的距离估计值，并且优化进化模型中核苷酸的替代速率参数，当核苷酸的替代速率参数优化到一定精度时，将最终获得的由最大似然法算得的距离估计值作为基于距离的进化树构建算法的输入。

虽然最大似然方法对于确定 quartet 的最大似然树可以在一个合理时间内得出，但是对于位点数目过大的序列来说，会花费更多的似然值计算时间[37]；并且对于每个 quartet 组合，每次都要优化三个不同拓扑结构的分支长度，计算其似然值，以选取具有最大似然值的拓扑结构估计进化距离，为了提高速度，对算法进行几点优化。

首先，经过观察，对于长度较长的序列，存在很多核苷酸分布相同的位点，即存在位点冗余。在进行最大似然值的计算之前，对序列进行预处理，将序列比对由 4 个扩展到全局，去除序列冗余简化序列，以在最大限度下保证不丢失序列信息。其具体过程是：设有 n 个 DNA 序列，针对每个位点，在所有序列之间进行观察比对，如果在位点 k，所有序列在该位点的核苷酸相同，则把它作为非信息位点舍弃；否则，只要存在两个或两个以上序列在该位点有不同的核苷酸，就保留该位点，并且对这样的位点进行统计，将具有相同核苷酸分布的位点简化为一个位点，将其重复的次数定义为该位点的权值，这样既最大限度地保留了有用的序列信息，又简化了序列，节省了很多对冗余位点的似然值的计算时间。

其次，由于要选取所有可能的 quartet 组合，并且对于每个 quartet 组合，每次都要优化三个不同拓扑结构的分支长度，计算其最大似然值，以选取最大似然值最大的拓扑结构估计进化距离，计算量也是较大的，所以对用最大似然法确定每个 qaurtet 的最终的拓扑结构的计算过程进行优化。在计算已知拓扑结构的最大似然值时，要优化各个分支长度，以使似然值达到最大，而这个过程需要多次迭代，需要大量的计算时间，所以优化计算的方法是：首先用最大似然法计算两两序列间距离，然后利用最小二乘法对一个 quartet 的三个拓扑结构进行分支长度估计，并利用该分支长度计算出近似的似然值；确定似然值最大的拓扑结构后，基于已有的分支长度进行最大似然值的优化计算，利用最后获得的分支长度对序列之间的距离进行估计。

最后，使用基于 quartet 的最大似然方法估计序列之间的距离，需要优

化很多棵四个类的树的似然值，所以，似然值的优化过程使用了一种简单的优化方法——Brent 优化方法，并不需要很多计算量[9]。在计算似然值最大的过程中，使用只具有一个参数的函数。这个方法是：在每个似然值优化阶段，定义三个值：a，b 和 M，并且满足 $a<M<b$，$f(a)<f(M)$ 和 $f(b)<f(M)$，我们要寻找的最优的值就是在 a 和 b 之间，并且 M 值为至今找到的具有最高函数值的点，连接 a、b 和 M 这三个点的只有一条抛物线，它的峰值对应的横坐标 s 定义了 M 的新的值。新的间隔$(a，b)$的定义依赖于 s 与 M 的关系，如果 $s<M$ 则新的间隔$(a，b)$等于$(a，M)$；如果 $s>M$，则$(a，b)$的值等于$(b，M)$。每一次迭代，值的搜索范围都会减小，当达到所要求的精确范围时，则迭代结束。

3.3 基于距离的建树方法

在基于距离的进化树重构算法中，邻接算法是利用进化距离矩阵构建进化树的方法中最为流行的。这个算法遵循最小进化原理，即选择分支长度之和最小的进化树作为最优树。之后出现了许多邻接算法的变形，试图通过设计能够找到最接近最小进化树的算法改善邻接算法。

Saitou and Imanishi 提出一个穷尽的搜索算法搜索最小进化树，但只适用于序列数较少的数据集合；Rzhetsky and Nei 通过对进化树的局部重新调整对邻接树附近的区域应用不同的策略来搜索最小进化树；还有许多人提出使用一个解靴带程序产生许多可供选择的拓扑结构[21]。最后，Kumar(1996)设计了一个启发式算法搜索树的空间。这些方法都能够产生一个短的进化树的集合，这个集合能够提供比邻接树更多的信息。而且，他们通常能够找到比邻接树更短的进化树。但是，计算机仿真实验显示这些方法覆盖真实进化树的能力并没有提高[38]。

基于距离的进化树构建算法中，首先，物种之间的距离估计值是重要的，另外还需要找到能够很好地处理这些输入值的进化树构建算法。下面先简要分析邻接算法及其优缺点，然后再对基于邻接算法在构建进化树的不足之处进行改进。

3.3.1 邻接法分析

邻接法是一种快速的聚类方法，与其他基于距离的进化树构建算法相比，邻接法在算法上相对较为复杂，它跟踪的是树上的节点而不是分类单

元。这种方法通过选择速率校正距离最小的两个类确定为待合并的类，然后将这两个类合并为一个类，并在相应的进化树上添加一个内部节点，将合并的两个类连接，作为两个类的父节点。然后，在距离矩阵中添加新的分类，根据原始距离矩阵，对每两个节点间的距离进行调整，即计算每个分类单元与新生成的类之间的进化距离，同时删除原来的两个分类。随后，重复上述过程。在每一次循环过程中，都有两个节点被一个新的父节点所取代，两个类被合并成一个新类。如果要构建有根树，整个循环直到只剩两个类为止；如果要构建无根树，则整个循环只需在剩余三个类停止。

邻接法的具体计算过程我们采用一个过程来解释，这个算法迭代的执行过程是：

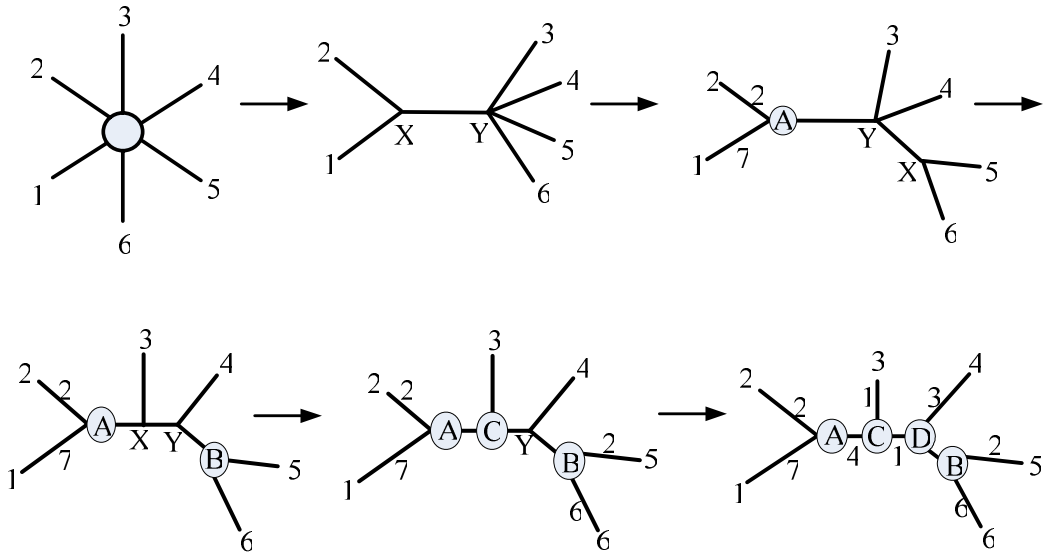

图 3-3 邻接法的计算过程图解

Figure 3-3 Graphic illustration of NJ computation

在计算分支长度时，邻接法与最小二乘法中的 Fitch-Margoliash 法是相似的，但它们在决定如何选择序列对时采用了不同的算法。邻接法不需要关于分子钟的假设，在进化速率可变的情况下是一种可靠的预测进化树的方法[39]。当所考虑的谱系间进化速率可变时，邻接法特别适用。当已知拓扑结构的树的枝长可以与进化变异的不同水平同时变化时，邻接法是一种高效的进化树构建算法，可以较快地构建进化树，也比较适合分析较大的数据集，并且同时给出拓扑结构和分支长度。

然而，邻接法还存在一些不足之处，还需要对其算法过程进行进一步的改善。

邻接法是基于最小进化原理提出的一种基于距离的进化树构建算法，如果核苷酸数目足够多，并且核苷酸替代数目的无偏估计被用作距离测度，则算法可以给出正确的拓扑结构，它只适合于处理可加的或接近可加的距离矩阵[40]。对于一棵进化树 T，假设每条分支的长度函数 l_T： $E(T)\to R^+$；树 T 中两个节点 u 和 v 之间的唯一路径由 $P_T(u,v)$表示。对于每一棵进化树 T 引入对应于树中叶子节点的一个距离函数：

$$D_T(a,b)=\sum_{e\in P_T(a,b)} l(e) \tag{3-1}$$

当一棵进化树 T 的距离矩阵 D 满足 $D=D_T$ 时，则称距离矩阵 D 是可加的；进化树 T 被称为实现了距离矩阵 D，并且它是唯一的，可以被表示为 $T(D)$。当距离矩阵 D 满足公式(3-2)时，称 D 为接近可加的。设 $\mu(T)$表示树 T 的最小的分支长度；$D(x,y)$表示物种 x 和物种 y 在距离矩阵中的进化距离值。

$$\left|D-D_T\right|_\infty < \mu(T)/2 \tag{3-2}$$

$$\left|D-D_T\right|_\infty = \max_{x,y\in n}\left|D(x,y)-D_T(x,y)\right| \tag{3-3}$$

所有可加的距离矩阵都可以构建出相同的拓扑结构，不考虑分支长度，进化树 T 对于可加的距离矩阵是唯一的。在这种情况下，邻接算法存在一个最理想的重构界限。实际上，大多数的距离矩阵都远不是接近可加的，更别谈及距离矩阵的可加性。所以，邻接算法的重构界限限制了邻接算法的应用范围，并不能准确处理所有的距离数值的输入。

另外，在每次迭代过程中，每聚合一对分类单元 i 和 j，都要对距离矩阵进行更新，去掉与被聚合的两个分类单元相关的行和列，加入一个新的行或列，并且要计算新的分类单元与其它分类单元之间的进化距离的估计值，如果这些新的估计值不够准确，这样会丢失距离矩阵中许多进化信息。在邻接法中，计算新的分类单元 u 与其它物种 k 之间的进化距离的公式如下：

$$d_{uk}=\frac{1}{2}d_{ik}+\frac{1}{2}d_{jk}-\frac{1}{2}d_{iu}-\frac{1}{2}d_{ju} \tag{3-4}$$

当 u 是 i 和 j 的父节点，k 是其他分类单元。可以看到，邻接法简单地使用不加权的平均公式来计算新的进化距离估计值，造成距离矩阵中的生物进化关系信息丢失，也导致邻接法构建进化树不够准确。

3.3.2 算法改进

由于邻接法存在的缺点，不能很好地处理由基于 quartet 的最大似然法产生的距离估计值，在基于邻接法的基础上，引入了进化距离的方差和协方差，并且利用距离的方差和协方差矩阵引入新的权值，能有效地处理进化距离中存在的噪声[41]，通过过滤影响基于最小进化原理的速率校正距离的取样噪音，使邻接法不再局限于处理理想界限以内的距离矩阵。

通常研究的 DNA 序列都代表同源物种，物种之间的进化距离一般都很小，当物种 i 与物种 j 之间的进化距离 d_{ij} 接近 0 时，方差和协方差矩阵其实是同一个[42]，这样引入的方差和协方差矩阵形同虚设，使获得的系统进化树并不可靠；而且，进化距离的方差和协方差矩阵的计算非常耗时。正如广义最小二乘法，并不需要精确的方差和协方差的计算[43]。

当距离矩阵中的距离估计值的准确性越差，就给他们越小的权值，使用该距离的方差表示，这种假设是计算新的距离估计值的基础。当距离的方差是可知的，则最优权值可由该方差求得。近似计算公式中，距离方差是和距离成正比的，公式如下：

$$v_{ij} \approx d_{ij} / s \tag{3-5}$$

式中 s——序列长度。

然而，这些距离的估计值之间并不是独立的。如果，物种 i 和 j 之间的路径和物种 k 和 l 之间的路径分享相同的分支时，则 d_{ij} 和 d_{kl} 之间是相关的[43]。在这种情况下，最合理的方法还要考虑距离估计值之间的协方差。设 u 和 v 代表树中节点 i、j、k 和 l 的祖先节点，则 u 和 v 是路径(i，j)和路径(k，l)的交叉部分的两个端点，d_{uv} 表示两个祖先节点间的进化距离。根据以上协方差的分析，其近似公式为：

$$\mathrm{cov}_{ij,kl} \approx d_{uv} / s \tag{3-6}$$

从公式(3-6)中我们可以看出，当路径(i，j)和(k，l)没有共同的分支时，$\mathrm{cov}_{ij,kl}$ 的值为空。

距离的方差和协方差并不需要准确的值，这种大约值是足够的[41]。上述这两个近似公式可以被应用到大多数进化距离估计值。

邻接法是一个迭代算法，每次迭代都要完成一次聚类，即将两个分类单元合并成一个新的分类单元，然后修改距离矩阵，去掉与被合并的两个分类

单元有关的行或列，并添加一个新的行或列，在这个过程中，如果新生成的进化距离不准确，很容易造成生物信息的丢失。在邻接法中，虽然使用了距离之间的平均趋异程度计算了新的分类单元与被聚合的两个分类单元之间的距离，即树中相应的分支长度，但是，对于新的进化距离的估计，只是通过公式(3-4)完成的，公式使用不加权的平均公式来计算新的进化距离估计值，使新的距离估计值不够准确，也导致算法构建拓扑结构不够准确。将公式(3-4)变成一般形式就是：

$$d_{uk} = \lambda d_{ik} + (1-\lambda)d_{jk} - \lambda d_{iu} - (1-\lambda)d_{ju} \tag{3-7}$$

所以，改进算法中，在对新的进化距离估计时，也考虑了相关距离之间的趋异程度，因为大多数距离矩阵都不是可加的，所以，不能用 $\lambda = 1/2$，这种情况只适用于可加的距离矩阵[44]；当距离矩阵不是可加时，路径(i，k)和路径(j，k)对共同拥有的路径(u，k)的贡献值是不同的，所以，需要计算(i，k)和路径(j，k)对共同拥有的路径(u，k)的贡献权值 λ。设 i 和 j 是待合并的分类单元，k 是其他分类单元。这个权值 λ 的计算公式如下：

$$\lambda = \frac{\sum_{k=3}^{r}\left(v_{ik} - \mathrm{cov}_{ik,jk}\right)}{\sum_{k=3}^{r}\left(v_{ik} + v_{jk} - 2\,\mathrm{cov}_{ik,jk}\right)} \tag{3-8}$$

λ 对于每对待聚合的分类单元都是不同的，所以每次聚合之前都要计算 λ。公式(3-7)的后两项是常数，只依赖于分支长度 d_{iu} 和 d_{ju}，而 d_{iu} 和 d_{ju} 已经通过公式(3-9a)、(3-9b)获得，不影响 λ 的值，对其起决定作用的是前两项 $\left(\lambda d_{ik} + (1-\lambda)d_{jk}\right)$，因此也对树的拓扑结构起决定作用。

$$d_{iu} = \frac{1}{2(m-2)}\left[(m-2)d_{ij} + S_i - S_j\right] \tag{3-9a}$$

$$d_{ju} = \frac{1}{2(m-2)}\left[(m-2)d_{ij} - S_i + S_j\right] \tag{3-9b}$$

随着距离矩阵的更新，方差矩阵和协方差矩阵都要随之发生变化，它们的变化是在原方差矩阵的基础上，新的距离 d_{uk} 的方差为 $v_{uk} = Var(d_{uk})$，将 d_{uk} 的计算公式(3-7)代入，因为变化的总是前两项，只需代入 $\left(\lambda d_{ik} + (1-\lambda)d_{jk}\right)$，获得公式(3-10)。

$$v_{uk} = \lambda^2 v_{ik} + (1-\lambda)^2 v_{jk} - 2\lambda(1-\lambda)\mathrm{cov}_{ik,jk} \tag{3-10}$$

相应的，协方差的计算公式可以按照协方差在树中的定义获得，$\mathrm{cov}_{uk,mk}=\mathrm{cov}(d_{uk},d_{mk})$，同样的，只需将$(\lambda d_{ik}+(1-\lambda)d_{jk})$代入，替换$d_{uk}$，获得协方差的计算公式(3-11)。

$$\mathrm{cov}_{uk,mk}=(v_{uk}+v_{mk}-v_{um})/2 \tag{3-11}$$

新的距离的估计方法更加精确了距离估计，再次使用统计方法得到相关距离的趋异程度，很大程度避免在距离矩阵更新时造成的信息丢失，使拓扑结构的构建更加准确。

3.4 基于距离的贪心算法的改进

邻接算法通过多次迭代完成整棵进化树的构建，每次迭代过程中，聚合能使树的所有分支长度之和最小的两个分类单元，生成新的树节点。它是一个贪心算法。

3.4.1 贪心特性分析

像邻接法一样，改进算法也是利用邻接法的速率校正距离实现最小进化原理。构建进化树之前，首先要建立一棵星状树，在进行选择最佳的待聚合的两个分类单元时，并不知道哪一对分类单元能使树的总的分支长度最小，因此，就把所有的分类对当作潜在的邻居，即待聚合的分类对，然后用与图 3-4 相似的拓扑结构来计算从第 i 个到第 j 个分类单元的分支长度，随后再选出有最小Q_{ij}的分类单元 i 和 j。

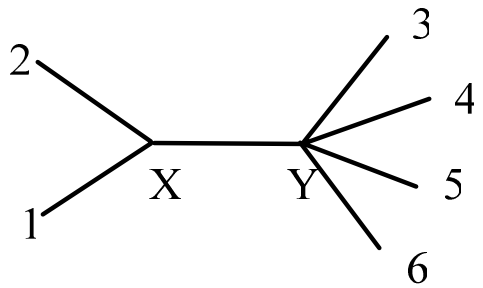

图 3-4 选取待聚合的分类单元

Figure 3-4 Selecting OTUS to combine

本例中，i 表示第 i 个外部分支节点，而 X，Y 是内部分支节点，L_{iX} 是

节点 i 和 X 间的分支长度估计。如上所述，要从这六个分类单元中找出能使所有分支长度之和最小的分类对，要考虑所有可能的分类对聚合后形成的树的所有分支长度之和。Q_{ij} 表示将分类单元 i 和 j 聚合后形成的树的所有分支长度的和值，则对于上图中的 Q_{12} 是由 $L_{1X}+L_{2X}$，L_{XY} 以及 $\sum_{i=3}^{n}L_{iY}$ 的总值决定的。这里，$L_{1X}+L_{2X}=d_{12}$ 而

$$L_{XY}=\frac{1}{2(n-2)}[\sum_{i=3}^{n}(d_{1i}+d_{2i})-(n-2)(L_{1X}+L_{2X})-2\sum_{I=3}^{n}d_{iY}] \tag{3-12}$$

$$\sum_{i=3}^{n}L_{iY}=\frac{1}{n-3}\sum_{3\le i<j}d_{ij} \tag{3-13}$$

$$Q_{12}=\frac{1}{2(n-2)}\sum_{i=3}^{n}(d_{1i}+d_{2i})+\frac{1}{2}d_{12}+\frac{1}{n-2}\sum_{3\le i<j}^{n}d_{ij} \tag{3-14}$$

如果 $S_1=\sum_{i=1}^{n}d_{1i}$，$S_2=\sum_{i=1}^{n}d_{2i}$ 且 $T=\sum_{i<j}^{n}d_{ij}$，则 Q_{12} 为：

$$Q_{12}=\frac{2T-S_1-S_2}{2(n-2)}+\frac{d_{12}}{2} \tag{3-15}$$

显然，Q_{12} 的值很容易的计算出来，如果用 i 和 j 代替上面的 1 和 2，又因为所有的 i、j 对的 T 值均相同。为了计算 Q_{ij} 的相对值，Q_{ij} 的计算公式可以简化为：

$$Q_{ij}=(n-2)d_{ij}-S_i-S_j \tag{3-16}$$

称 Q_{ij} 为速率校正距离，也称转移距离。

贪心算法在每次迭代过程中，遍历 n 个对象的 C_n^2 个组合的 Q_{ij}，选择 Q_{ij} 最小的一对分类单元的组合，即最近邻对，这样能够保证遵循最小进化原理，使得到的树的所有分支长度之和最小。

只有当距离矩阵是可加的，使用最小进化原理构建的进化树是最优的，然而大多数距离矩阵远不是可加的，这样得到的具有最小的分支长度之和的拓扑结构并不一定是真实拓扑结构的无偏估计。尽管前面改进算法利用序列之间距离的方差和协方差，有效地改进了进化树的重构，但是算法并没有改进距离矩阵的可加性以及这种贪心搜索的特性，正是由于这种贪心特性，使得邻接法不能总是找到最小进化树，只能找到拓扑结构和最小进化树相似的较短的进化树[6]。

3.4.2 贪心特性的改进

为了改进这种贪心特性，我们引入了“neighbor”的概念，将其应用在改进算法中。这样既可以快速构建系统进化树，而且还把算法的平均时间复杂度降低。

首先，“neighbor”的定义是这样的，选取序号为 i、j 两个节点，按照公式(3-17)、(3-18)，分别计算与其他物种之间的速率校正距离 Q_{ik} 和 Q_{jk}，并且分别找出速率校正距离最小的分类单元组合 Q_{ip} 和 Q_{jq}，如果 i 与 j 之间的速率校正距离 Q_{ij} 满足 $Q_{ij}=Q_{ip}$ 和 $Q_{ij}=Q_{jq}$，即 $i=q$ 和 $j=p$，则定义 i 与 j 是一对“neighbor”。

$$Q_{ip}=\min\{Q_{ik}\mid k=0\sim n,k\neq i\} \tag{3-17}$$

$$Q_{jq}=\min\{Q_{jk}\mid k=0\sim n,k\neq j\} \tag{3-18}$$

通过实验发现，最近邻对的贪心搜索不仅浪费很多的搜索时间，而且并不是全局最优的，即速率校正距离最小的配对并不一定是在真实的进化树中进化距离最近的[45]，在距离矩阵中，还存在一些距离矩阵支持较少的聚合，这些聚合通常不满足速率校正距离最小。所以改进算法中，每次迭代选择只要满足“neighbor”判断标准的分类单元对进行聚合。这种方法改进了在邻接算法中的贪心选择特性。因为在距离矩阵中，可能存在不止一个“neighbor”对，最近邻对也是一对“neighbor”，且是所有存在的“neighbor”对中速率转移距离最小的，而对选取两个对象进行对“neighbor”的判断，过程的算法复杂度不会超过 $\mathrm{O}(n\lg n)$。

与改进算法相比，邻接算法每次迭代都要进行 C_n^2 次查找，然而最后，算法只聚合一对分类单元组合，时间复杂度为 $O(n^2)$，浪费较多的搜索时间。邻接法在确定节点之间是否为邻居时支持具有最大一致的聚合标准，这个标准是通过搜索最小的速率转移距离来执行的。而引入“neighbor”后改进算法以相等的概率聚合每一对“neighbor”，增大了由距离矩阵支持较少的潜在聚合的几率；而邻接算法每次只聚合速率校正距离最小的“neighbor”对，可以看出，改进算法不是像邻接法一样，在每一次聚合中，都使用最大程度的贪心思想。正是由于邻接法的贯穿整个算法的贪心思想导致整个体系的偏差[46]，而改进算法虽然也存在产生体系偏差的可能

性，但是这种可能性却很大程度减少了这种体系偏差对真实的进化关系的影响：由相同的距离矩阵所支持的所有的节点聚合操作具有相等的概率，而邻接法却因为总是选择速率校正距离最小的最近邻，完全地排除了由距离矩阵支持较少的潜在聚合[46]。

因此，通过随机选择“neighbor”对能够较好地减小选择最近邻时可能导致的偏差，并且有效地提高覆盖真实进化树的能力。对于每次迭代过程，至少存在一对“neighbor”，所以，在改进算法中使用随机选择“neighbor”对来代替贪心选择算法，如果每次随机选取两个对象进行对“neighbor”的选取，过程的算法复杂度是$\mathrm{O}(n\lg n)$[46]，为了提高搜索速度，我们可以将这一过程进行推广，随机选取 m 个对象，同时进行“neighbor”对的选取，相当于 m 个“neighbor”对的选取过程并行进行，则这个算法的时间复杂度不会超过$\mathrm{O}(n\log_2 n)$。在本论文算法的实现中，我们选取了$m=3$。实验结果证明改进了贪心特性后的算法能够更好地提高构建的进化树质量。

3.5 完整算法介绍

改进算法主要是在邻接法的基础上进行改进，其中改进算法主要分三部分：

(1) 距离估计方法的改进 利用局部优化的最大似然方法对进化距离进行估计，改进了传统的进化距离估计方法的确定，即每次只对两个序列进行观察，计算其信息位点数，并且对于所有序列，没有考虑单个位点上的多重替代，所以它们的距离估计是不准确的，通常会低估了真实的进化距离；在最大似然方法中，考虑了所有位点的核苷酸，把序列比对由两个序列比对扩大到多重序列比对，这样可观察到的序列信息大大增多。所以，改进算法利用基于quartet的最大似然法来优化核苷酸替代速率，精确两个对象之间的进化距离。

(2) 利用距离方法构建进化树 由于产生的距离矩阵并不满足可加性，引入了进化距离的方差和协方差，并且利用距离的方差和协方差矩阵引入新的权值，能有效地处理进化距离中存在的噪声[41]，通过过滤影响基于最小进化原理的速率校正距离的取样噪音，使邻接法不再局限于处理理想界限以内的距离矩阵，并使算法能够构建更加精确的拓扑结构；

(3) 贪心聚合的改进 正是由于邻接法的贯穿整个算法的贪心思想导致整个体系的偏差[46]，而改进算法引入了“neighbor”的概念，改进了邻接法

聚合速率校正距离最小的“neighbor”对，而是聚合每一对“neighbor”。增大了由距离矩阵支持较少的潜在聚合的几率；改进算法虽然也可能存在产生体系偏差的可能性，但是这种可能性却很大程度减少了这种体系偏差对真实的进化关系的影响。

进化树重构算法的算法复杂度依赖于数据集合中的序列数 n，其算法复杂度的分析可以分为两个阶段：(1) 进化距离矩阵的估计：邻接法在这一阶段的时间复杂度为 $\mathrm{O}(n^2)$，因为在进行进化距离估计的过程中，每两个序列之间的进化距离都会被计算；改进算法在这一阶段的算法复杂度为 $\mathrm{O}(n^4)$，因为该算法需要计算所有可能的 quartet 组合，故较邻接法要高；(2) 基于距离矩阵的进化树的构建：邻接法的算法复杂度为 $\mathrm{O}(n^3)$，而改进算法由于只需寻找一对满足“neighbor”定义的两个分类单元进行聚合，其时间复杂度为 $\mathrm{O}(n^2 \log_2 n)$。由此可以看出，改进算法的总体时间复杂度较邻接法高，所以，该算法是以消耗时间的代价换取拓扑结构的准确性。

算法 3-1 a) 基于 quartet 的进化树构建算法

Algorithm 3-1 a) Quartet-based phylogenetic tree construction

输入：n 个 DNA 序列

输出：n 个 DNA 序列的进化树

1. 初始化 $n \times n$ 的进化距离矩阵 (δ_{ij})；

1.1 对于 n 个 DNA 序列：

1.1.1 对 n 个 DNA 序列进行预处理，即进行 n 重序列比对，统计位点信息，转化成新的序列模式；

1.1.2 用距离估计方法粗略估计一个进化距离矩阵 (d_{ij}) 中每两个物种之间的距离；

1.1.3 从 quartet 中取样，优化进化模型的参数，利用对两两序列使用最大似然法估计其距离，并更新距离矩阵 (d_{ij})。

1.2 利用 (d_{ij}) 计算每个可能的 quartet 的具有最大似然值的拓扑结构，并累加到距离矩阵 (δ_{ij}) 中相应位置，最后对 (δ_{ij}) 取平均值；

2. 初始化其他参数

2.1 初始化对象数目：$r \leftarrow n$；

2.2 初始化方差矩阵：$(v_{ij}) \leftarrow (\delta_{ij})$ 其中 $v_{ij} \approx \frac{1}{s} d_{ij}$，这里 s 是序列长度；

算法 3-1 b) 基于 quartet 的进化树构建算法

Algorithm 3-1 b) Quartet-based phylogenetic tree construction

3. 当对象的数目 r 大于 3 时：

3.1 随机选取三个不同的对象 a、b 和 c；

3.2 对 a、b 和 c 分别按照公式 $Q_{ij}=(r-2)\delta_{ij}-S_i-S_j$，计算 $Q_{ip}=\min\{Q_{ik}\mid k=0\sim n,k\neq i\}$，如果存在“neighbor”对，设为 i、j，则将 i、j 作为要聚合的两个对象；否则，转 3.1 步；

3.3 确定 i 和 j 与聚合点 u 之间的支长，使用公式 $\delta_{iu}=\frac{1}{2}\left(\delta_{ij}+\frac{S_i-S_j}{(r-2)}\right)$；

3.4 确定 λ，使用公式

$$\lambda=\frac{\sum_{k=3}^{r}\left(v_{ik}-\mathrm{cov}_{ik,jk}\right)}{\sum_{k=3}^{r}\left(v_{ik}+v_{jk}-2\,\mathrm{cov}_{ik,jk}\right)}$$

其中 $\mathrm{cov}_{ik,jk}=\frac{1}{2}\left(v_{ik}+v_{jk}-v_{ij}\right)$；

3.5 更新距离矩阵 $\left(\delta_{ij}\right)$，计算聚合点 u 与其他对象 k 之间的距离利用公式 $\delta_{uk}=\lambda\delta_{ik}+(1-\lambda)\delta_{jk}-\lambda\delta_{iu}-(1-\lambda)\delta_{ju}$；

3.6 更新方差矩阵 $\left(v_{ij}\right)$，利用公式 $v_{uk}=\lambda v_{ik}+(1-\lambda)v_{jk}-\lambda(1-\lambda)v_{ij}$；

4. 使用 3.3 步中的公式，计算最后三个对象之间的分支长度；

5. 输出树；

3.6 实验及结果分析

3.6.1 实验介绍

由于古生物化石几乎没有留下有意义的古生物大分子信息，我们只能从现代生存着的物种的生物大分子获得信息，并据此推断大分子进化史，建立系统进化树，然而真实的物种序列是很少的，并且我们缺乏客观的评价标准来评价这些树的正确性，所以我们采用了评价建树算法中最常用的方法——计算机模拟法。

计算机模拟法是首先设立一个树的模型作为标准的进化树，按照这个树的拓扑结构，通过模拟生物的进化过程，生成需要处理的序列。这个模拟过程将被重复很多次，就会生成许多组序列。然后按照不同的算法，在这些序列的基础上构建出进化树。如果算法构建的进化树与标准进化树的拓扑结构差异越小，则说明其准确性也就越高。

实验主要分为三个部分：(1) 数据集合；(2) 拓扑结构准确性；(3) 计算时间。下面，首先介绍实验中采用的数据集合，然后介绍如何测量算法构建的进化树的拓扑结构准确性及计算时间测试。

3.6.1.1 数据集合 实验测试遵循一个曾被 Kumar(1996)、Gascuel(1997)和 Ranwez(2001)在一个相似结构中使用过的协议[47]。一个进化树的重构方法的效率依赖于模型树的形状、进化速率和分子时钟是否保持不变等[48]，所以采用了六棵具有 12 个类的模型树，如图 3-5。

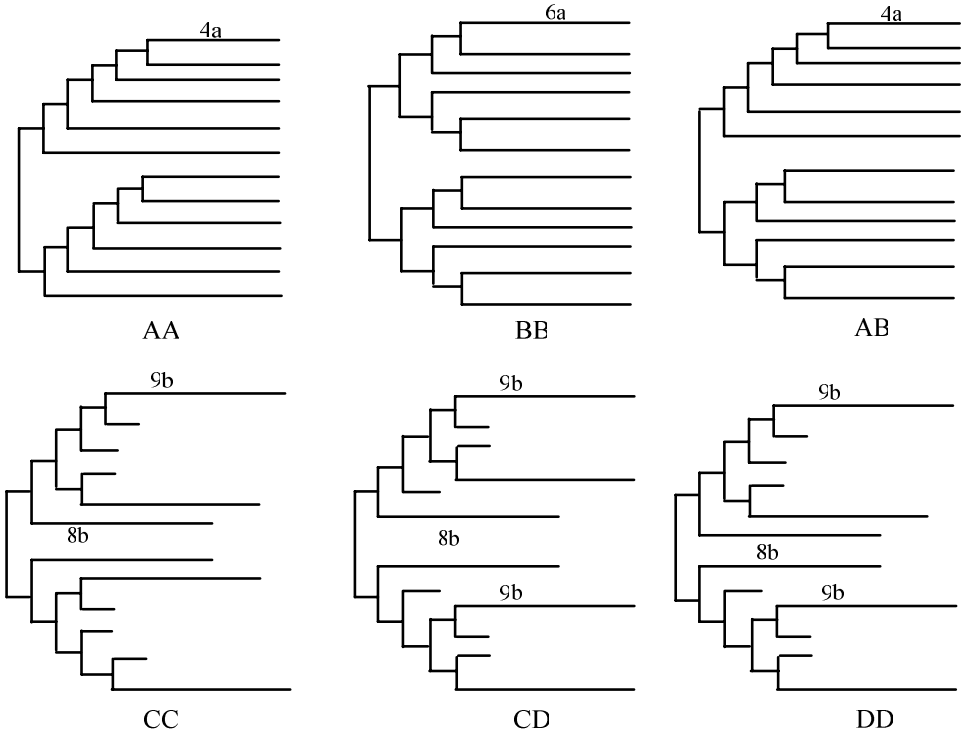

图 3-5 计算机模拟法中的标准进化树

Figure 3-5 Standard phylogenetic trees in computer simulations

前三棵树(AA，BB，AB)满足分子时钟假设，而后三棵树(CC，DD，CD)显示替代速率在世系之间是变化的。对于每一个内部分支是一个单位长度(用 a 表示)，外部分支长度是 a 与 b 的倍数。因此，对于这些标准进化

树，数据集合是在三种进化环境下产生的：

(1) 较慢的进化速率：当 a=0.00625 和 b=0.005 时，每个位点的最大替换率约为 0.1($MD \approx 0.1$)。

(2) 适中的进化速率：当 a=0.0185 和 b=0.015 时，每个位点的最大替换率约为 0.3($MD \approx 0.3$)。

(3) 较快的进化速率：当 a=0.0625 和 b=0.05 时，每个位点的最大替换率约为 1.0($MD \approx 1.0$)。

对于每棵树的每种进化速率，又包含 100 组序列长度为 300 的数据和 100 组序列长度为 600 的数据。

一棵模型树，用 T 表示，是使用被 Kuhner 和 Felsenstein (1994)描述的随机物种形成过程生成的一棵进化树，这个过程与通常的 Yule-Harding 关于树的描述是相应的[49]。这些仿真数据是使用一个被 Guindon and Gascuel (2002)开发的软件获得的，使用 SEQGEN_1.06[50]产生相应的 DNA 序列。对于每一棵模型树 T，这些序列都是通过沿着树 T 进行一个进化过程仿真模型产生的。这些数据可以在 http://atgc.lirmm.fr/datasets/benchmarks.html 被获得。

3.6.1.2 拓扑结构准确性 拓扑结构的准确性依赖于每两个拓扑结构之间的距离测度，要计算这个距离测度，首先需要对进化树的拓扑结构使用标准的符号表示，进化树的拓扑结构的信息在计算机程序中常常用一组嵌套的圆括号表示，称为 Newick 格式(Newick format)[51]。

例如，图 3-6 的 Newick 格式表示是：((1，(2，3))，(4，5))；图 3-7 的其中一种 Newick 表示是((1，2)，5，(3，4))。

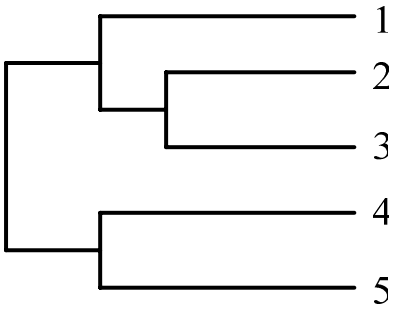

图 3-6 一个 5 序列的有根树

Figure 3-6 A rooted tree for five taxa

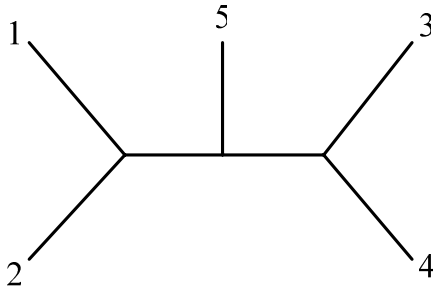

图 3-7 一个 5 序列的无根树

Figure 3-7 An unrooted tree for five taxa

在这种标准的树的符号表示形式下，利用拓扑距离计算方法对重建树与标准树之间的拓扑距离构建一个统一的距离测度。常用的有两种拓扑距离计算方法：分支得分距离(branch score distance of kuhner and felsenstein (1994))和 Robinson and Foulds 距离(symmetric difference of Robinson and Foulds(1981)，也叫 RF 距离)。分支得分距离使用了树的分支长度，并且只有当树在所有的分支都有长度时才能被计算出；RF 距离并不使用分支长度信息，只与树的拓扑结构有关。

任何一种拓扑结构距离都要有一个直观的统计上的解释。这些距离的计算考虑在两棵树中存在的所有内部分支，每一个分支都会将序列截断，分成两个组，一组和该分支的一个端点相连，一组和它的另一个端点相连。例如一棵树((A，C)，(D，(B，E)))，有两个内部分支，其中一个分支导致一种截断{A，C | B， D， E}，另一个分支导致另一种截断{A，C，D | B，E}。具有相同物种集合的另一棵树为(((A，D)，C)，(B，E))，它的内部分支导致的两种截断为{A，C，D | B，E}和{A，D | B，C，E}。注意，其他分支都是外部分支，导致的截断只能把一个物种和其他物种分开，所以它们在所有树中导致的截断都是一样的，在计算拓扑结构距离时，并不考虑外部分支导致的截断。所以上述两棵树，在考虑所有内部分支的截断后，有两个不同的截断，则这两棵树的拓扑结构距离为 2。这就是 RF 距离的计算方法，它只考虑在两棵树中，有多少个截断只在一棵树中有，而另一棵树中没有的数目。

而在分支得分距离的计算过程中，树中存在的每一个截断都有一个分支长度与之相关联。例如一棵树为(((A：0.1，D：0.25)：0.05，C：0.01)：0.2，(B：0.3，E：0.8)：0.2)，每个分支导致的截断及其相关联的分支长度列表为：

{A | B，C，D，E}　0.1
{D | A，B，C，E}　0.25
{A，D | B，C，E}　0.05
{C | A，B，D，E}　0.01
{A，D，C | B，E}　0.4
{B | A，C，D，E}　0.3
{E | A，B，C，D}　0.8

对于两棵具有相同的序列集合的进化树，考虑所有可能的这些列表，然后将只在一棵树中存在的截断的相关联的分支长度的平方进行累加，得到的值即为两棵树的拓扑距离。如果两棵树中都没有这样的分支，则这两棵树的拓扑结构距离为 0。

本实验中，使用的是 RF 距离测度，拓扑结构距离是用 RF 距离为测度来衡量的，是通过一个软件实现的 RF 距离的计算，即 Phylip3.66 中的 treedist.exe。对于每一组数据集合，统计由进化树构建算法得到的那些与真实的进化树之间的 RF 距离，然后求出平均值，即测试结果与真实树之间的拓扑结构距离。这个数值越低，则证明算法越准确。

为了更好地评价改进算法构建进化树的准确性，不仅用原算法(即经典邻接法，Saitou and Nei，1987)和改进算法对数据进行测试，还在相同的数据集合上测试了基于邻接法的改进算法中最为经典的 BIONJ(Gascuel 1997)，最后将改进后算法的测试结果分别与原算法和 BIONJ 算法的测试结果进行对比。实验结果如表 3-1、表 3-2、表 3-3、表 3-4、表 3-5、表 3-6 所示。

3.6.1.3 计算时间　为了测试改进算法与原算法以及其他算法的实际运行速度，选择了两组数据集合来进行实验，它们分别是基于 24 个类的和基于 96 个类的，数据是在核苷酸替代速率分别为“慢速”、“中等”、“快速”三种环境下产生的，其中对于 24 类集合，分别是 0.03、0.06 和 0.15；对于 96 类集合，分别是 0.02、0.04 和 0.10[52]。这样，实验就是在六组数据(http://www.lirmm.fr/w3ifa/MAAS/)上进行拓扑结构和运行时间测试。

测试环境：PC 机一台，处理器 1.5G，内存 512M，WinXP 操作系统。

3.6.2 实验结果与分析

首先，下列六个表是三种算法在三个不同的进化环境下基于六棵标准进

化树的测试结果。

表 3-1 邻接法和改进算法对树型 AA 的测试结果

Table 3-1 Experiment results of NJ and improved algorithm for tree AA

	算法	300bp	600bp
MD≈0.1	邻接法	11.34	9.76
	BIONJ	5.62	4.1
	改进算法	3.62	2.36
MD≈0.3	邻接法	9.32	8.34
	BIONJ	4.46	3.7
	改进算法	2.6	2.02
MD≈1.0	邻接法	8.58	8.1
	BIONJ	5.84	4.12
	改进算法	2.34	2.04

表 3-2 邻接法和改进算法对树型 BB 的测试结果

Table 3-2 Experiment results of NJ and improved algorithm for tree BB

	算法	300bp	600bp
MD≈0.1	邻接法	12.9	10.96
	BIONJ	6.7	4.46
	改进算法	3.46	2.02
MD≈0.3	邻接法	10.52	9.08
	BIONJ	4.98	3.86
	改进算法	2.4	2.0
MD≈1.0	邻接法	9.68	8.72
	BIONJ	6.84	4.92
	改进算法	2.5	1.48

表 3-3 邻接法和改进算法对树型 AB 的测试结果

Table 3-3 Experiment results of NJ and improved algorithm for tree AB

	算法	300bp	600bp
MD≈0.1	邻接法	14.96	13.22
	BIONJ	7.9	5.84
	改进算法	3.6	1.74
MD≈0.3	邻接法	13.22	11.62
	BIONJ	6.5	5.12
	改进算法	2.58	1.48
MD≈1.0	邻接法	13.46	11.56
	BIONJ	7.96	6.28
	改进算法	3.1	1.1

表 3-4 邻接法和改进算法对树型 CC 的测试结果

Table 3-4 Experiment results of NJ and improved algorithm for tree CC

	算法	300bp	600bp
MD≈0.1	邻接法	18.0	18.0
	BIONJ	9.66	10.32
	改进算法	4.8	5.3
MD≈0.3	邻接法	18.0	18.0
	BIONJ	11.1	9.76
	改进算法	7.44	4.44
MD≈1.0	邻接法	18.0	18.0
	BIONJ	10.08	9.94
	改进算法	3.72	3.32

表 3-5 邻接法和改进算法对树型 DD 的测试结果

Table 3-5 Experiment results of NJ and improved algorithm for tree DD

	算法	300bp	600bp
MD≈0.1	邻接法	18.0	18.0
	BIONJ	9.9	9.68
	改进算法	6.64	6.2
MD≈0.3	邻接法	18.0	18.0
	BIONJ	9.9	8.98
	改进算法	4.88	4.1
MD≈1.0	邻接法	18.0	18.0
	BIONJ	9.9	8.88
	改进算法	4.88	2.92

表 3-6 邻接法和改进算法对树型 CD 的测试结果

Table 3-6 Experiment results of NJ and improved algorithm for tree CD

	算法	300bp	600bp
MD≈0.1	邻接法	18.0	18.0
	BIONJ	9.52	10.06
	改进算法	6.1	5.32
MD≈0.3	邻接法	18.0	18.0
	BIONJ	9.86	9.54
	改进算法	5.32	5.04
MD≈1.0	邻接法	18.0	18.0
	BIONJ	9.88	9.44
	改进算法	3.26	3.62

从表 3-1、表 3-2、表 3-3、表 3-4、表 3-5、表 3-6 的结果中不难发现，改进的算法要比原有算法性能上明显提高，得到的拓扑结构更加准确，因为

使用了局部优化的最大似然法获得的更加准确进化距离估计，并利用该距离的方差和协方差矩阵，使算法对新的进化距离估计更为精确，有效地解决了邻接法对不可加的距离矩阵估计的偏差，并且通过“neighbor”减小选择最近邻时可能导致的偏差，改进了邻接法贪心特性，使算法不总是寻找分支长度之和最短的进化树，有效地提高了覆盖真实进化树的能力，使这种聚类构树方法更加准确。

从表 3-7 中可以看出，结果是分别在 24 类和 96 类的数据集合上测试得到的，为了将两种数据上的结果进行对比，拓扑距离为两棵拓扑结构中不同分支数目的比例，即拓扑距离的值为 RF 距离在内部分支数中的比例。由于改进算法考虑了所有可能的 quartet 的最大似然结构，所以时间耗费较大，但是从表 3-7 中可以看出，改进算法可以在一个可接受的时间内进行进化树的重构。综上，实验证明改进算法已达到了预期的目标。

表 3-7 运行时间对比

Table 3-7 Computing time comparison

	算法	n=24		n=96	
		拓扑距离	运行时间(秒)	拓扑距离	运行时间(秒)
慢速	邻接法	0.3214	1	0.4280	1
	BIONJ	0.1310	1	0.2086	1
	改进算法	0.0595	3	0.1634	30
中等	邻接法	0.2857	1	0.4086	1
	BIONJ	0.1190	1	0.1505	1
	改进算法	0.0476	5	0.1054	42
快速	邻接法	0.2381	1	0.4172	1
	BIONJ	0.1090	1	0.1572	1
	改进算法	0.0348	12	0.0924	56

3.7 本章小结

本章主要介绍了基于邻接算法的三部分改进：(1) 进化距离的估计方法；(2) 基于距离的进化树的构建方法；(3) 对邻接法贪心特性的改进。分别用了三小节对邻接法的不足之处进行分析，并对该算法进行描述。然后，对整个算法流程进行具体介绍。最后，采用大量的测试数据，对改进算法进行验证，并与邻接法和基于邻接法的经典改进算法 BIONJ 进行对比，给出实验结果。

第4章　进化树构建系统设计与实现

4.1　系统描述

为使本文提出的基于距离的进化树构建算法得以应用，设计了一个进化树构建系统，它完整地实现了从数据输入到结果输出、结果保存、进化树显示等全过程，并集成到计算分子生物学平台上。

系统主要功能包括：邻接法构树、最大似然法构树、改进算法构树和画树程序。系统平台界面如图 4-1 所示。

图 4-1　系统平台界面

Figure 4-1 System interface

4.2 数据形式

4.2.1 输入数据

系统的输入文件和树的输出文件使用统一的标准格式。首先介绍一下输入数据格式——DNA 序列文件：DNA 序列文件可以含有一组序列，也可以含有多组序列；对于每组序列，第一行的两个字段分别是序列个数和位点长度，以下每一行记录一条 DNA 序列信息，直到序列全部显示。其中，每一行的起始部分(与后续部分由多个空格分隔)是该 DNA 序列代表的物种名称(不超过 10 个字符)，后续部分对应该物种的 DNA 序列。其具体格式如图 4-2。

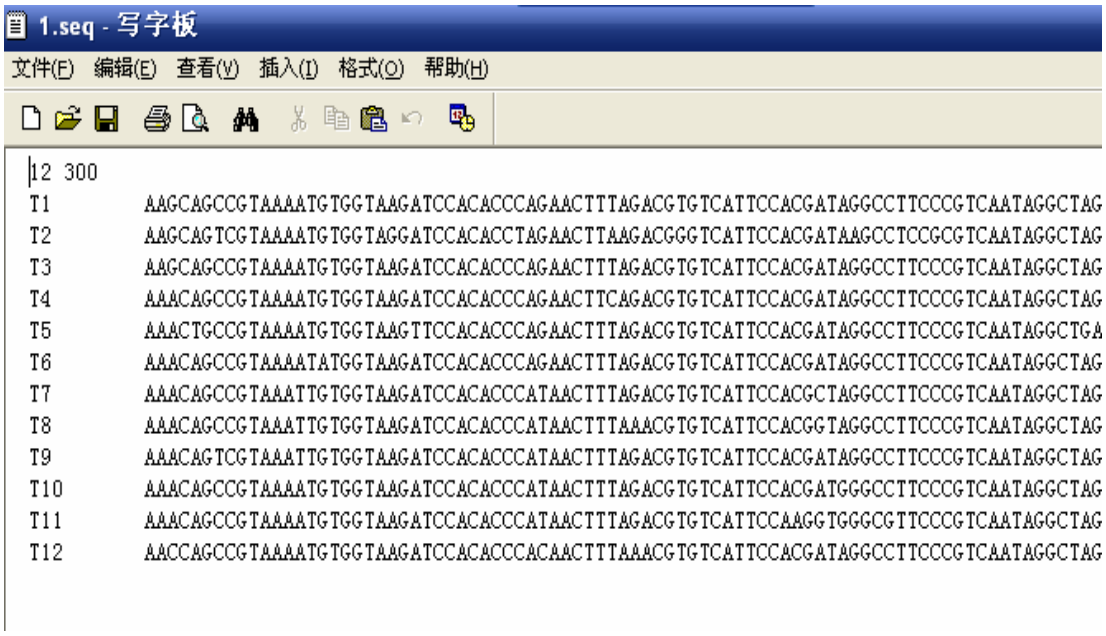

图 4-2 输入文件格式

Figure 4-2 The input file format

4.2.2 输出数据

输出文件的数据是采用 Newick 格式的进化树，采用嵌套的括号形式表示树的拓扑结构以及分支对应的长度，分支与分支长度之间是以“:”进行连接；每棵进化树的最后必须以“;”结束。具体格式见图 4-3。

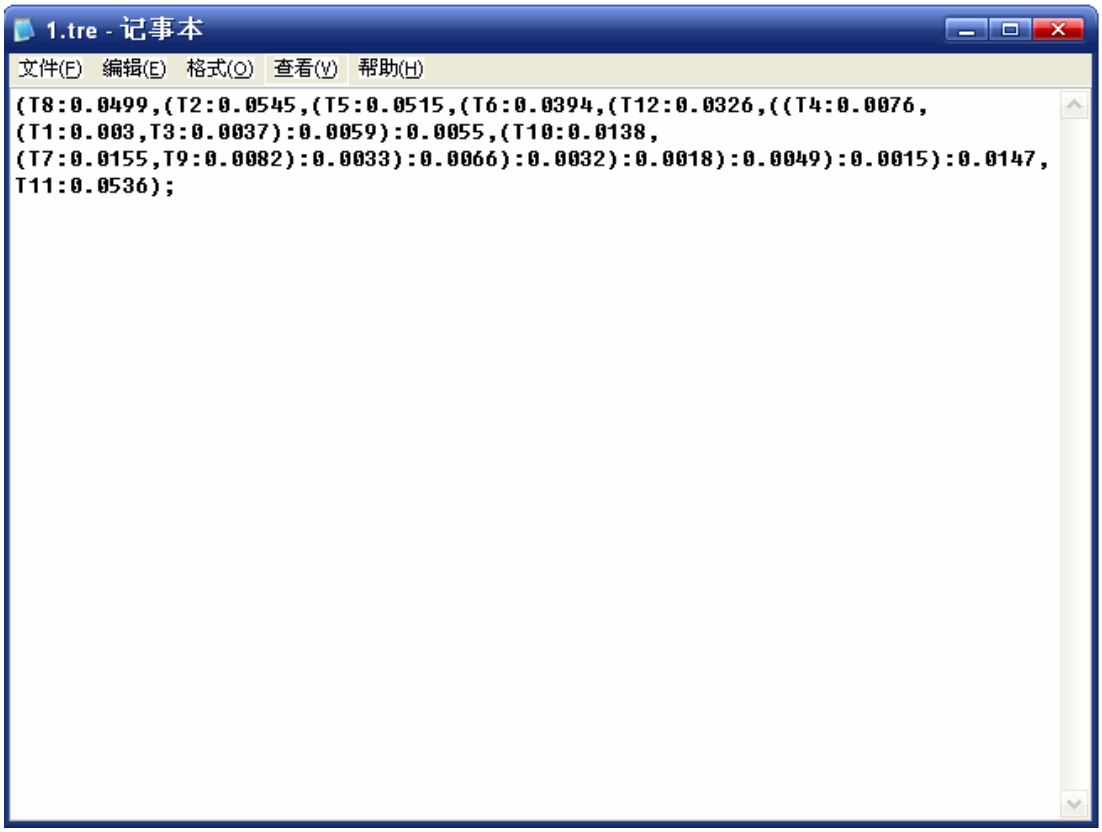

图 4-3 树的输出形式

Figure 4-3 The output format of trees

4.3 主要功能模块

根据进化树构建算法的需求，将系统划分为四个主要功能模块：

(1) 邻接算法模块：集成了经典邻接法和基于邻接法的改进算法BIONJ。用户可以任选其一进行操作，同时可以比较它们的输出结果。

(2) 最大似然法模块：实现了最大似然法的构建进化树过程。同时，用户可以自定义迭代次数，使得系统更加灵活。

(3) 改进算法模块：实现了使用改进算法构建进化树的操作。

(4) 画树程序模块：一个独立的可视化显示进化树结构模块，将进化树以图形方式直观地呈现给用户。

另外，前三个模块均能完成 DNA 序列文件输入、输入显示、结果输出、进化树可视化显示等操作。

功能框架如图 4-4 所示。

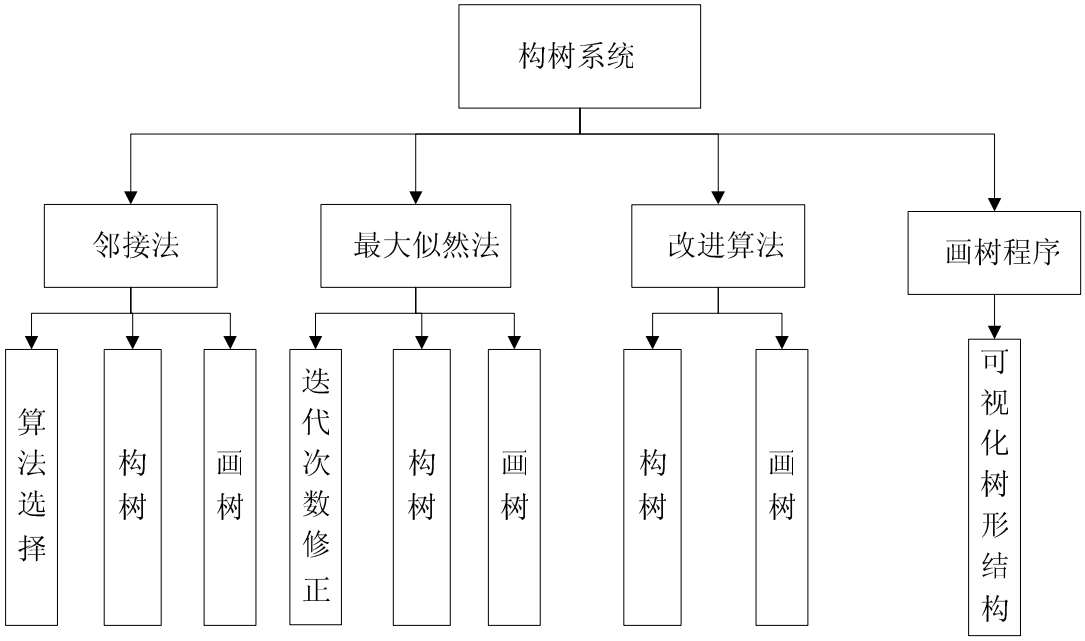

图 4-4 系统功能框架

Figure 4-4 Function Module of system

其中，构树子功能模块使用相应算法对输入的 DNA 序列进行建树操作，画树子功能模块将所得树以二维图形方式显示出来。

4.4 操作流程

操作流程主要包括以下四个部分：邻接算法模块、最大似然法模块、改进算法模块和画图程序模块。

4.4.1 邻接法

邻接算法模块集成了经典邻接法和基于邻接法的改进算法 BIONJ，可完成 DNA 序列文件输入、输入显示、结果输出、进化树显示等功能，并可将两个算法的输出结果进行对比。

操作步骤如下：

第一步，点击“浏览”按钮，输入 DNA 序列文件，上方文本框显示文件内容，中间文本框显示文件内容及其路径，如图 4-5 所示。

图 4-5 输入文件操作

Figure 4-5 Input file operation

第二步，选择构树方法，然后点击“构树”按钮，下方文本框显示树的输出形式，“保存”按钮完成进化树的输出数据存储操作。例如：选择“NJ(经典)”构树，过程及结果如图 4-6 所示。

图 4-6 对选中的 DNA 序列文件进行进化树构建操作

Figure 4-6 Constructing phylogenetic trees on the selected DNA sequence file

第三步，进化树构建完成之后，点击“画树”按钮，弹出该进化树的二维显示图形。在该模块中，可使用两个算法构建相同的 DNA 序列，并可将得到的进化树分别进行可视化显示，便于拓扑结构的比较。图 4-7 显示的是邻接算法对输入文件 1.seq 进行构建进化树操作后的图形显示。

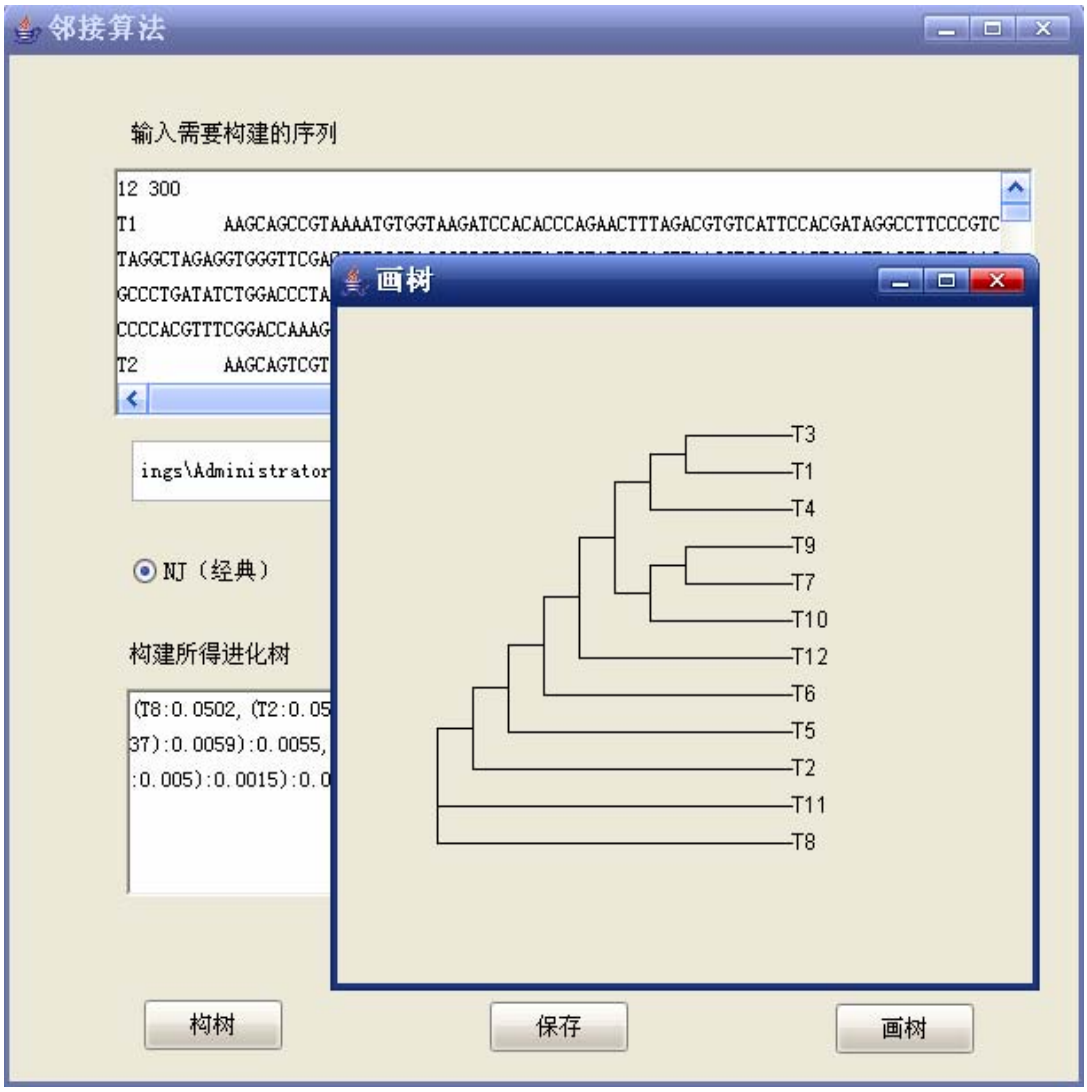

图 4-7 显示由经典邻接法构建的进化树的图形

Figure 4-7 Display the phylogenetic tree produced by neighbor-joining algorithm

另外，对已选择的 DNA 序列文件使用 BIONJ 算法构建进化树的操作过程同 NJ 算法。

4.4.2 最大似然法

最大似然法模块实现了最大似然法构建进化树的操作，其中，对似然值的计算是使用 PHYLIP3.66 中的 BRENT 方法。在此模块中，可根据需要设置每个支长迭代的次数，完成 DNA 序列文件输入、输入显示、结果输出、

进化树显示等功能。

操作过程如下：

选择 DNA 序列文件，然后设置迭代次数(默认值为“1”)，然后点击“构树”按钮，得到的结果如图 4-8 所示，点击“画树”按钮，得到树的二维结构图。

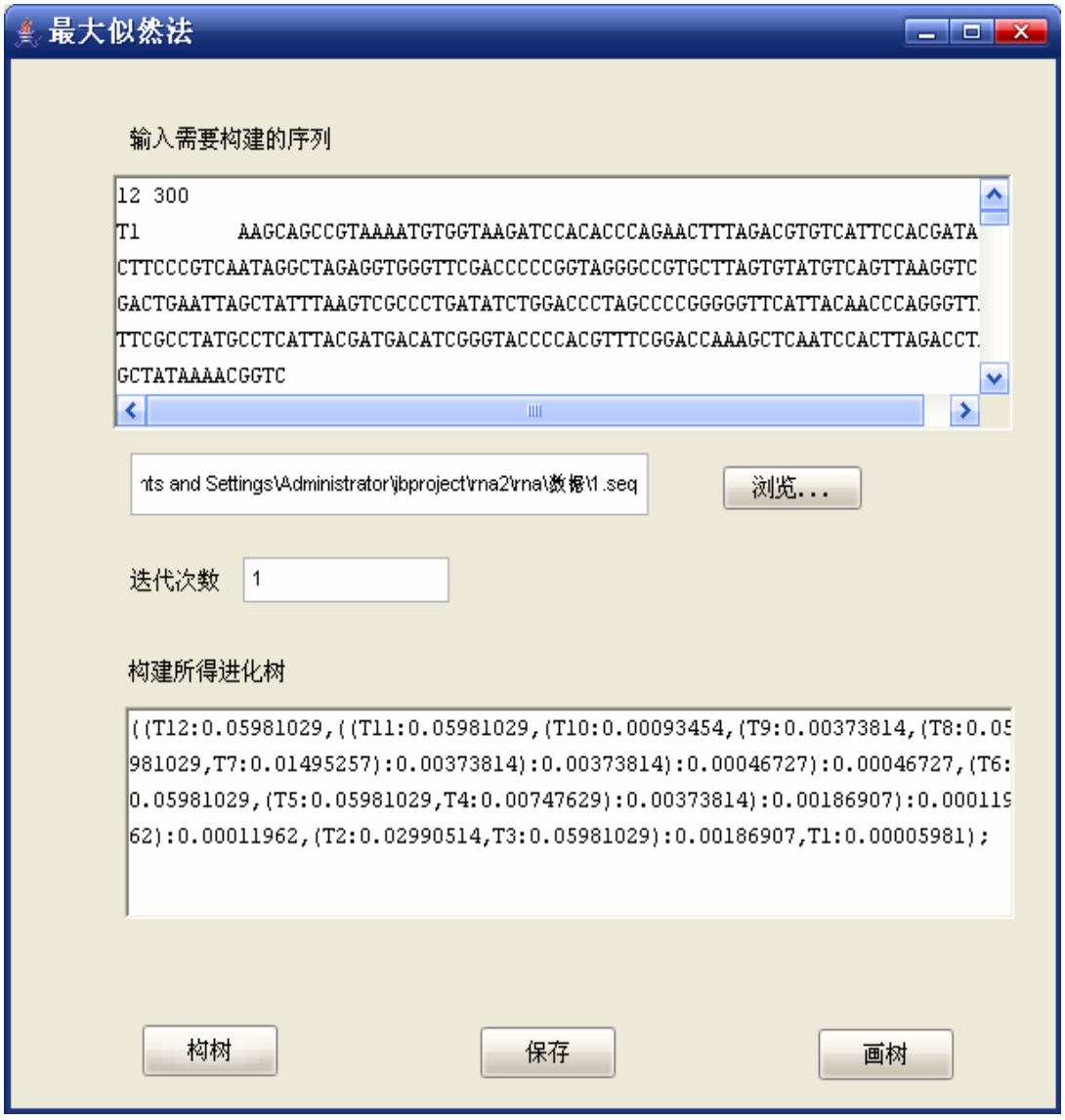

图 4-8 使用最大似然法构建进化树操作

Figure 4-8 Constructing phylogenetic tree by maximum likelihood algorithm

4.4.3 改进算法

改进算法模块实现了改进算法构建进化树操作，界面如图 4-9 所示。操作过程如下：

选择 DNA 序列文件，然后点击“构树”按钮得到计算结果，点击“画树”按钮得到树的二维结构图。

图 4-9 使用改进算法构建进化树操作

Figure 4-9 Constructing phylogenetic tree operation by improved algorithm

4.4.4 画树程序

画树程序模块是一个相对独立的模块，它可以直接对输入的树文件进行二维图形显示，既可以显示无分支长度的拓扑结构，也可显示带分支长度的拓扑结构，输入树文件必须以“;”结束。主要操作是树文件的输出和结果保存，方便用户查看已有树文件的树形结构，见图 4-10 和图 4-11。

4.5 技术方案

本系统使用 Java 语言开发，具有跨平台特性，可以移植到不同的操作系统上。

开发平台：Java2 platform standard edition；

开发工具：Borland JBuilderX Enterprise edition；

主要运行环境：JDK1.5.0。

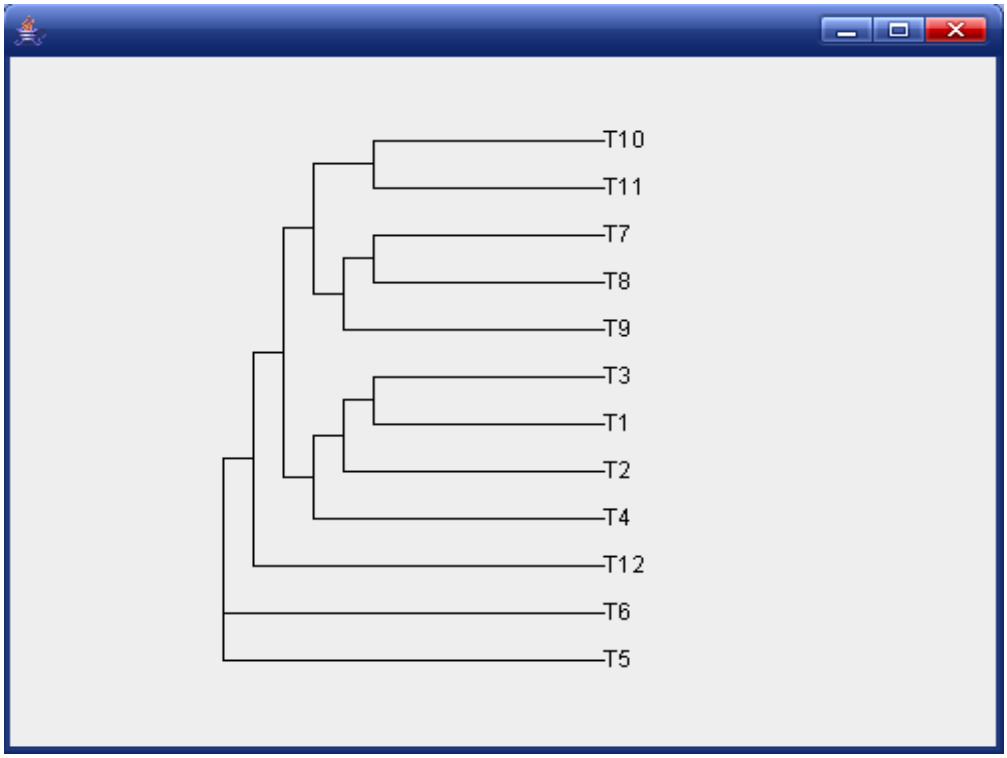

图 4-10 无分支长度的拓扑结构

Figure 4-10 Topology without branch length

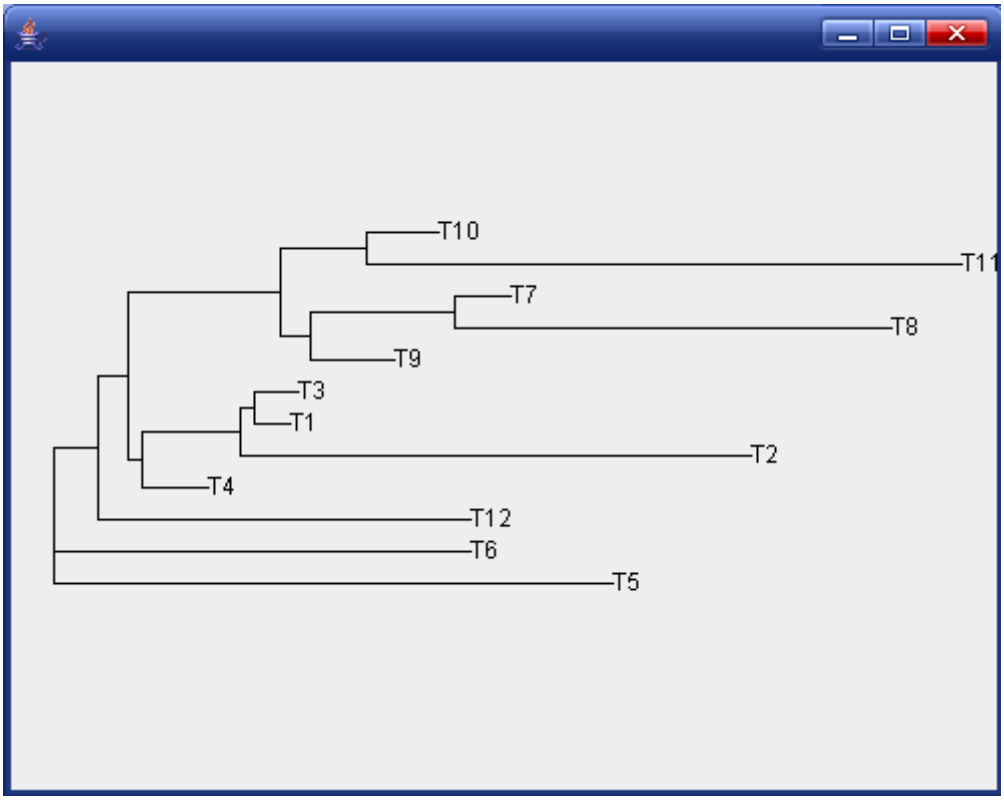

图 4-11 带分支长度的拓扑结构

Figure 4-11 Topology with branch length

4.6 本章小结

本章主要介绍了一个系统平台，实现了用不同方法构建进化树的平台，集成了经典构树方法和改进算法。使用该系统可以方便地选择任何输入文件，进行进化树的重构，并可对算法进行比较和进化树的二维图形显示。

结　论

本文主要研究的是基于距离的进化树构建算法，算法主要分为两个阶段：(1) 使用进化距离估计方法完成对距离矩阵的初始化；(2) 基于进化距离矩阵，进行进化树的重构工作。

在对进化距离估计方面，主要进行了两方面的研究：一是对传统的进化距离估计方法的研究，在这些方法中，每次只选择两个序列进行观察，计算其信息位点数，对于所有序列，并没有考虑单个位点上的多重替代，因而造成它们对距离估计不准确，在从特征数据到距离数据的转化过程中，会丢失大量有用信息，所以这种方法通常会低估真实的进化距离。二是对基于 quartet 的最大似然法的距离估计研究，在最大似然方法中，把序列比对由两个序列比对扩大到多重序列比对，而且对每个可能的 quartet 组合，利用最小二乘法确定具有最大似然值的拓扑结构，以对每两个序列之间的距离进行估计，似然函数的计算是基于拓扑结构的所有位点的似然值乘积，其每个参数为核苷酸的替代数，所以核苷酸替代数的估计过程(序列之间距离值的估计过程)就转化为似然函数的最大化过程，使进化距离估计更加准确。

在基于距离的进化树构建方面，主要是在邻接法的基础上进行改进，改进工作分为两部分：一是针对最大似然方法获得的距离矩阵的不可加性，引入了距离的方差和协方差，改进了邻接法只适合于处理可加的或接近可加的距离矩阵的不足，同时改进了其使用不加权的平均公式来计算新的进化距离估计，采用加权的计算公式更新距离矩阵，使算法构建的拓扑结构更加准确。二是改进了邻接法的贪心特性，因为邻接法每次只聚合速率校正距离最小的两个分类单元，所以容易导致整个体系产生偏差，实验证明速率校正距离最小的配对并不一定是在真实的进化树中进化距离最近的，而改进算法每次聚合速率校正距离满足“neighbor”的两个分类单元，这种方法在迭代过程选择了一些由距离矩阵支持较少的潜在聚合，这些聚合通常不满足速率校正距离最小，很大程度减少了这种体系偏差对真实进化关系的影响。

但是，本文仍存在着一些不足，改进算法相对于 NJ 法时间消耗较大，因此，对于如何实现更快速地处理较大规模数据的算法，还需要进一步的研究。另外，这种方法能够产生由邻接法构建的进化树的父集，可以搜索到更多准确的进化树，可将其应用于最优树搜索算法中。

参考文献

1 李涛, 赖旭龙, 钟扬. 利用 DNA 序列构建系统树的方法. 遗传 (Beijing). 2004, 26(2): 205~210

2 谭严芳, 金人超. 一种基于 NJ 的高效构建系统进化树算法. 计算机工程与应用. 2004, 21: 84~97

3 钟扬, 王莉, 张亮. 生物信息学. 高等教育出版社, 2003: 212~248

4 Luay Nakhleh. Phylogenetic Networks. The University of Texas at Austin. 2004, 5

5 Saitou N, Nei M. The Neighbor-Joining Method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 1987, 4(4): 406~425

6 Saitou, Imanishi. Relative Efficiencies of the Fitch-Margoliash, Maximum-Parsimony, Maximum-Likelihood, Minimum-Evolution, and Neighbor-Joining methods of Phylogenetic Reconstructions in Obtaining the correct tree. Mol. Biol. Evol. 1989, 6: 514~525

7 Korbinian Strimmer, Arndt von Haeseler. Quartet Puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 1996, 13(7): 964~969

8 Katherine St. John, Tandy Warnow, Bernard M. E. Moret, Lisa Vawter. Performance Study of Phylogenetic Methods: (unweighted) quartet methods and neighbor-joining. Journal of Algorithms. 2003, 48: 173~193

9 Vincent Ranwez, Olivier Gascuel. Improvement of distance-based phylogenetic method by a local maximum likelihood approach using triplets. Mol. Biol. Evol. 2002, 19(11): 1952~1963

10 L. Bao Zhong. Constructing Molecular Evolutionary Trees. Zool Research, 1993, 14(2): 186~193

11 Jukes, T. H. and C. R. Candor. Evolution of Protein Molecules. In Mammalian protein metabolism(H. N. Munro, ed.), Academic, New York. 1969, pp. 21~32

12 Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. 1980, 16: 111~120

13 Tajima, F. and M. Nei. Estimation of evolutionary distance between nucleotide sequences. Mol. Biol. Evol. 1984, 17: 115~120
14 Wolstenholme, D. R. Animal mitochondrial DNA: Structure and evolution. In international review of cytology: Mitochondrial genomes (D. R. Wolstenholme and K. W. Jeon, eds.), 1992: 173~216
15 Tamura, K. The rate and pattern of nucleotide substitution in Drosophila mitochondrial DNA. Mol. Biol. Evol. 1992, 9: 814~825
16 Tamura, K. and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol. Biol. Evol. 1993, 10: 512~526
17 C. Cotta, P. Moscato. Inferring phylogenetic trees using evolutionary algorithms. Lecture Notes in Computer Science. 2002, 249: 720~729
18 J. Felsenstein. Phylogenies from molecular sequences: inference and reliability. Annuals Rev. Genetics. 1988, 7(22): 521~565
19 Li Jing-Yan. The pdaric method for constructing molecular evolutionary trees from sequences data. Zool Research, 1992, 13 (4): 387~396
20 S. Holmes. Statistics for phylogenetic trees. Theor. Pop. Biol. 2003, 63: 17~32
21 Rzhetsky and Nei. Theoretical foundation of the minimum-evolution method of phylogenetic inference. Mol. Biol. Evol. 1993, 10: 1073~1095
22 Kidd, k.k., and L .A. Sgaramella-Zonta. Phylogenetic analysis: concepts and methods. Am. J. Hum. Genet. 1971, 23: 235~252
23 Studier, J. A. and K. J. Keppler. A note on the neighbor-joining method of Saitou and Nei. Mol. Biol. Evol. 1988, 5: 729~731
24 J. Felsenstein. Inferring phylogenies from protein sequences by parsimony, distance and likelihood methods. Methods Enzyme. 1996, 2(66): 418~27
25 A. Moilanen. Searching for most parsimonious tree with simulated evolutionary optimisation. Clad. 1999, 15: 39~50
26 Z. Yang, N. Goldman, A. Friday. Comparison of models for nucleotide substution used in maximum likelihood phylogenetic estimation. Biol. 1994, 11(2): 316~224
27 Pearson W. R., Robins G., Zhang T. Generalized neighbor-joining: more reliable phylogenetic tree reconstruction. Mol. Biol. Evol, 1999, 16(6):

806～816
28 Li W. H. Evolutionary change of restriction cleavage sites and phylogenetic inference. Genetics, 1986, 113: 187~213
29 C. B. Congdon. Phylogenetic trees using evolutionary search: Initial progress in extending gaphyl to work with genetic data. In Congress on Evolutionary Computation, Canberra, Australia. 2003: 56~81
30 Holder M., Lewis P. O. Phylogeny estimation: traditional and Bayesian approaches. Nature, 2003, 4: 275~284
31 K. Katoh, K. Kuma. Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. J. Mol. 2001, 53: 477~484
32 Kuhner, M. K. and J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. 1994, 11: 459~468
33 Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 1981, 17: 368~376
34 Le Sy Vinh, Arndt von Haeseler. Shortest triplet clustering: reconstructing large phylogenies using representative sets. BMC Bioinformatics. 2005
35 P.L. Erd″os, M. Steel, L. Székély, T. Warnow, A few logs suffice to build almost all trees—I, Random Structures Algorithms. 1997, 14: 153~184
36 G.J. Olsen, H. Matsuda, R. Hagstrom, R. Overbeek, fastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood, Comput. Appl. Biosci. 1994, 10: 41~48
37 Peter G. Foster. The idiot's guide to the zen of likelihood in a nutshell in seven days for dummies, unleashed. 2001, 7
38 Kumar's. A stepwise algorithm for finding minimum evolution trees. Mol. Biol. Evol. 1996, 13: 584~593
39 A. Edwards. Estimation of the branch points of a branching diffusion processes. J. Royal Stat. Soc. 1970, 32: 155~174
40 Isaac Elias and Jens Lagergren. Fast Neighbor Joining. L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, 2005: 1263~1274
41 Olivier Gascuel. BIONJ: an improved Version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. 1997, 14(7): 685~695
42 Rzhetsky and Nei. Statistical properties of the ordinary least-squares，

generalized least-squares and minimum-evolution methods for phylogenetic inference. J. Mol. Evol. 1992, 35: 367~375

43 Bulmer, M. Use of the method of generalized least squares in reconstruction phylogenies from sequence data. Mol. Biol. Evol. 1991, 15: 1346~1359

44 Felsenstein, J. Estimation of hominoid phylogeny from a DNA hybridization data set. J. Mol. Evol. 1987, 26: 123~131

45 P. Arndt, C. Burge. DNA sequence evolution with neighbor-dependent mutation. Proceedings of the Sixth Annual International Conference on Computational Biology, 2002, 2: 32~38

46 Jason Evans, Luke Sheneman, James Foster. Relaxed neighbor joining: a fast distance-based phylogenetic tree construction method. 2001

47 Ranwez, V. and O. Gascuel. Quartet-based phylogenetic inference: improvements and limits. Mol. Biol. Evol. 2001, 18: 1103~1116

48 M. K. Kuhner, J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. 1994, 11: 459~468

49 Harding, E.F. The probabilities of rooted-tree shapes generated by random bifurcation. Adv. Appl. Probab. 1971, 3: 44~77

50 Rambaut, A. and N. C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 1997, 13: 235~238.

51 D. Baxevanis, B. Francis. Bioinformatics: A practical guide to the analysis of genes and proteins. John Wiley & Sons. 1998, 8: 45~62

52 Richard Desper, Olivier Gascuel. Fast and Accurate Phylogeny Reconstruction Algorithms Based on the Minimum-Evolution Principle. Journal of Computational Biology. 2002, 9(5): 687~705

哈尔滨工业大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《基于距离的进化树构建算法研究》，是本人在导师指导下，在哈尔滨工业大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字：　　日期：　　年　　月　　日

哈尔滨工业大学硕士学位论文使用授权书

《基于距离的进化树构建算法研究》系本人在哈尔滨工业大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨工业大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅，同意学校将论文加入《中国优秀博硕士学位论文全文数据库》和编入《中国知识资源总库》。本人授权哈尔滨工业大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

作者签名：　　日期：　　年　月　日

导师签名：　　日期：　　年　月　日

哈尔滨工业大学硕士学位涉密论文管理

根据《哈尔滨工业大学关于国家秘密载体保密管理的规定》，毕业论文答辩必须由导师进行保密初审，外寄论文由科研处复审。涉密毕业论文，由学生按学校规定的统一程序在导师指导下填报密级和保密期限。

本学位论文属于　　保密□，在　　年解密后适用本授权书

不保密□

(请在以上相应方框内打“√”)

作者签名：　　日期：　　年　月　日

导师签名：　　日期：　　年　月　日

致　谢

本论文的完成，首先要感谢我的导师郭茂祖教授，他的细心的指导和帮助使我顺利完成论文。感谢郭老师两年来对我学习和生活上的关怀和鼓励。他严于律己、宽以待人的工作作风，谨慎严谨的治学态度一直激励着我。研究工作的顺利开展、研究成果的取得都倾注老师的心血，在此谨向悉心培养、教育、帮助我的恩师表示衷心的感谢。

感谢刘扬老师在撰写论文期间给予的诸多帮助，感谢他不辞辛劳地帮我修改论文，并提出许多宝贵意见，衷心祝愿刘老师工作顺利，事业有成。

特别感谢李建伏师姐一直以来对我无私的指导和帮助，师姐严谨治学的态度让我受益匪浅，无论在学习上还是在生活上师姐一直对我的照顾我会铭记在心，衷心地祝愿她在以后的道路上事业有成，家庭幸福。

感谢富楠楠、张涛涛、张志田、唐武、李燕芬、罗贵存等同学在课题研究期间对我的帮助和支持。感谢他们在我遇到困难的时候，尽力地帮助我。他们敢于面对困难，解决困难的精神是我学习的榜样。祝愿他们每个人一路走好。

我要深深地感谢我的父母和男友，他们对我学业上的支持、在精神上的鼓励、在生活上的关心，给了我克服困难的勇气和不断进取的力量，给我最坚定的支持。祝福父母身体健康、安享晚年，也希望男友能够顺利和我一起踏上工作岗位。

最后，感谢在求学路上曾经教导过我的师长、同学以及曾经帮助过我的人，因为有你们的协助才能使我顺利完成各阶段的学业，以及能让我顺利的读完研究生阶段并完成本论文的撰写，谢谢你们。

感谢所有关心我的每一个人，祝福他们健康快乐！