硕 士 学 位 论 文

MASTER'S DISSERTATION

论文题目 RNA二级结构形状-碱基对距离及半监督聚类算法研究

作者姓名 李佳慧

学位类别 工程硕士

指导教师 王常武教授

2019年5月

中图分类号：TP302.7　　学校代码：10216
UDC：004　　密级：公开

工程硕士学位论文

（应用研究型）

RNA 二级结构形状-碱基对距离及半监督聚类算法研究

硕士研究生：李佳慧
导　　师：王常武教授
副　导　师：李永强高级工程师
申请学位：工程硕士
工程领域：计算机技术
所在单位：信息科学与工程学院
答辩日期：2019 年 5 月
授予学位单位：燕山大学

A Dissertation in Computer Technology

RNA SECONDARY STRUCTURE SHAPE-BASE PAIR DISTANCE AND SEMI-SUPERVISED CLUSTERING ALGORITHM

by Li Jiahui

Supervisor: Professor Wang Changwu

Yanshan University

May, 2019

燕山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《RNA 二级结构形状-碱基对距离及半监督聚类算法研究》，是本人在导师指导下，在燕山大学攻读硕士学位期间独立进行研究工作所取得的成果。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

燕山大学硕士学位论文使用授权书

《RNA 二级结构形状-碱基对距离及半监督聚类算法研究》系本人在燕山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燕山大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解燕山大学关于保存、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。本人授权燕山大学，可以采用影印、缩印或其它复制手段保存论文，可以公布论文的全部或部分内容。

保密□，在　　年解密后适用本授权书。

本学位论文属于

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

摘 要

RNA 是生命体内重要的大分子之一，不仅在遗传信息翻译中起决定作用，还具有酶的催化、细胞调节、病毒遗传信息携带等功能。RNA 功能多样性的关键在于 RNA 空间结构，而 RNA 空间结构的构成由 RNA 二级结构决定。真实 RNA 二级结构存在于高于最小自由能一定阈值范围内的 RNA 二级结构集合中。因此，通过计算 RNA 二级结构间距离区别 RNA 二级结构差异，并使用高效的聚类算法对 RNA 二级结构集合进行划分筛选代表结构，对真实 RNA 结构预测有重要意义。本文对 RNA 二级结构距离计算算法和 RNA 二级结构聚类算法进行研究，内容如下。

首先，针对已有 RNA 二级结构距离计算算法计算依据单一，误差性大的问题，提出 RNA 二级结构形状-碱基对距离计算算法。算法第一步是计算 RNA 二级结构的形状距离，将 RNA 二级结构抽象为带符号的有序树，通过转换、删除的树编辑操作来计算形状距离。第二步通过归一化思想计算形状距离和碱基对的平均分。第三步对 Rsd-bp 算法进行多进程优化，实现计算效率的提升。

其次，针对传统算法采用随机选择的方式初始化中心点，易陷入局部最优解的问题，提出半监督的 RNA 二级结构 k-medoids 算法，根据 Rsd-bp 算法计算获得 RNA 二级结构的距离矩阵和约束集合，并对约束集合进行预处理，获得监督信息。利用监督信息进行中心点初始化和数据划分，并改进 k-medoids 算法中心点更新规则，缩小中心点更新时需要查找数据的范围。

最后，通过两组对比实验分别验证 Rsd-bp 算法和 SS-medoids^{+}算法的可行性。第一组实验测试了 Rsd-bp 算法计算 RNA 二级结构距离的区分能力和计算效率。第二组实验对 RNA 二级结构集合进行 SS-medoids^{+}聚类实验。通过计算聚类评价指标，从聚类质量和时间两方面验证 SS-medoids^{+}算法有效性性。

关键词：RNA 二级结构；树模型；距离计算；半监督聚类；k-medoids；Rsd-bp

Abstract

RNA one of the indispensable large molecules in a living body, not only plays a decisive role in the translation of genetic information, but also boasts the functions of enzyme catalysis, cell regulation, viral genetic information carrying, etc. RNA spatial structure is critical to RNA functional diversity as RNAs with different spatial structures differ in function. RNA secondary structure decides the formation of RNA spatial structure. Real RNA secondary structure exists in the set of RNA secondary structures within an energy range above the minimum free energy. Therefore, it is of great significance to predict the real RNA structure by calculating the distance between RNA secondary structures, then dividing the set of RNA secondary structures by clustering algorithm, and screening the representative structures of each cluster for further research.In this paper, RNA secondary structure distance calculation algorithm and RNA secondary structure clustering algorithm are studied.

First, this paper presents Rsd-bp, a RNA shape - base pair distance calculation algorithm, to solve the problems of bigger error and lacking calculation basis diversification in existing algorithm for the calculation of distance between RNA secondary structures. The first step of this algorithm is to work out the shape-distance of RNA secondary structure RNA secondary structure is abstracted as a signed ordered tree and shape-distance is calculated through editing operations like conversion or deleting. The second step combines the shape-distance calculation and base pair calculation, computing the average score of the results of these two algorithms using normalization. Computation efficiency gets lifted in the third step by conducting multi-process optimization on Rsd-bp algorithm.

Second, traditional algorithm initializes medoids through random selection, which may only result in local optimal solution. Therefore, semi-supervised k-medoids algorithm for RNA secondary structure is put forward, which computes the distance matrix and the constraint set of RNA secondary structure based on Rsd-bp algorithm and preprocesses the constraint set to acquire the supervision information. Medoid initialization and data

classification are performed with the help of supervision information. Medoid updating rules of k-medoids are improved to narrow down the data searching scope while updating the medoids.

Finally, the feasibility of the two algorithms, Rsd-bp and SS-medoids^{+}, is validated through two comparative experiments. Distinguishing capability and computing efficiency of Rsd-bp in calculating RNA secondary structure distance is tested in the first experiment, while SS-medoids^{+} clustering experiment is conducted on the set of RNA secondary structures in the second experiment. SS-medoids^{+} is validated in terms of clustering quality and time by computing the clustering evaluation index.

Keywords: RNA secondary structure; tree model; distance calculation; semi-supervised clustering; k-medoids;Rsd-bp

目 录

第 1 章 绪　论

1.1 课题背景与研究意义

核糖核酸(RNA)是生命必须的四大分子之一。分子生物学中心法则(Francis Crick.1957)指出，“遗传信息从 DNA 传递给 RNA，再从 RNA 传递给蛋白质”[1]。表明 RNA 对于遗传信息传递和蛋白质合成有着重要的作用。除此之外，随着研究的深入，发现 RNA 具有酶的催化作用、细胞调节作用、病毒的遗传信息携带等功能。更有相关临床信息表明，某些 RNA 中的缺陷[2]或 RNA 的调节与人类疾病[3]、基因进化[4]等都有内在联系。

RNA 功能多样性的关键在于其空间结构[5]，且 RNA 空间结构形成具有层次性[6]。Sarah Woodson 实验室的结构/功能实验[7]据这一原理，提出了 RNA 的折叠模式。首先使单链 RNA 根据碱基互补原则实现自身折叠，形成带有互补区(茎区)的二维元件；然后由茎区连接的二维结构元件提供支架指导三级结构形成，三级结构受外部阳离子浓度变化影响形成活性位点构成最终 RNA 空间结构。在折叠过程中三级结构的形成不会引起二级结构的变化，所以可以独立描述这两个过程[8]，也就是说 RNA 二级结构的形成与三级结构无关。

细胞内环境影响因子复杂，在细胞内模拟合成 RNA 的研究有待成熟[9]，使用核磁共振和 X 射线晶体衍射等试验方法预测 RNA 空间结构，成本昂贵，十分耗时[10]。开发数学和计算方法来预测 RNA 结构成为 RNA 空间结构深入研究的关键。由于无法直接使用一级序列进行三级结构的预测，且 RNA 分子功能由 RNA 二级结构确定，所以研究热点集中于 RNA 序列的真实二级结构预测。

在过去的研究中，RNA 二级结构的自由能可加性是假设的，基于最小自由能模型预测 RNA 二级结构时，自由能参数的轻微差异就会对计算结果产生很大的影响。最新的研究结果表明真实 RNA 二级结构存在高于其自由能最小值 5~10%范围内的次优二级结构集合内，而次优二级结构的数目随 RNA 序列长度增长呈指数增长。随着非编码 RNA(ncRNA)家族数量的增加和识别[11]，使已知 RNA 序列的长度范围不断扩大，而长序列 RNA 对应的二级结构数量级十分庞大。如何对数目众多的次优二级结构集合进行分析研究，对 RNA 真实二级结构的预测具有重要意义。

1.2 国内外研究现状

目前 RNA 二级结构的研究算法可以分为两类思想，基于单序列研究算法和基于多序列研究算法。

1.2.1 基于单序列研究算法

基于单序列的研究算法以 RNA 的单个序列为输入对 RNA 二级结构进行预测，根据预测的能量模型不同，可分为基于能量最小的 RNA 二级结构预测算法和基于能量次优的 RNA 二级结构预测算法。

(1)基于能量最小的 RNA 二级结构预测算法

该类算法根据热力学模型方法，假定生物体内的环境为恒定，自由能最小时 RNA 二级结构最为稳定。代表算法有：基于动态规划算法预测 RNA 二级结构、基于遗传和模拟退火算法预测 RNA 二级结构、基于 hopfiled 的神经网络算法预测 RNA 二级结构。

基于动态规划 RNA 二级结构预测算法，依据热力学模型，通过组合子问题求解 RNA 二级结构的最优解。1978 年 Nussinov 基于动态规划算法首次提出最大碱基配对算法[12]，简单的认为单链 RNA 自身折叠使其碱基达到最大互补配对时，RNA 二级结构具有的自由能最小，结构最稳定。1984 年 Zuker 等人提出基于动态规划算法的最小自由能算法[13]。该算法对最大碱基对模型提出了修改，认为 RNA 二级结构的不同结构单元应赋以不同能量值。在一定温度下 RNA 分子通过构象调整达到某种热力学平衡时，其自由能最小，形成结构最稳定。在当时生物探究的背景下，这一预测模型的得到了普遍认可。但其预测精度较低，研究学者对这一算法进行了改进。如，Zuker(1989)提出了一种方法来确定自由能在最优值指定范围内结构的所有碱基对[14]。McCaskill(1990)重新定义动态规划算法来计算平衡分区函数，构建碱基对概率精确计算模型[15]。谭光明等(2006)通过并行计算对 Zuker 算法进行进一步优化，缩短了计算时间和计算精度[16]。

基于遗传和模拟退火算法预测 RNA 二级结构，通常基于能量最小结构最稳定的原则通过搜索最优子结构，求解 RNA 二级结构的组合优化问题。Wiese 等人[17]提出遗传算法预测 RNA 二级结构，该算法采用逐步选择的方式选择最优结构，过程类似于自然选择的淘汰过程。首先根据 RNA 序列中可能配对的碱基构建茎区池，然后利

用螺旋区堆积算法构造初始二级结构，最后采用自由能作为构造群体的适应标准，经过突变操作对结构中的茎区进行替换[18]。遵循能量越低结构越稳定的原则，迭代选择直到得出最优结果。而模拟退火算法的核心在于模仿热力学中高温固体或金属溶液冷却结晶的退火过程。退火过程遵循自由能减小定律，当自由能达到最小值时系统达到平衡态。具体的优化求解过程一般为根据抽样法则产生解决问题的解集，通过状态概率在温度下的变化来控制当前解，逐步向最优解转移的迭代过程。模拟退火作为一种迭代搜索优化算法，可以获得更高的稳定性。Tsang H 等人[19]提出 SARNA-Predict 算法采用改进的模拟退火算法作为搜索引擎，结合了基于排列的 RNA 结构编码和不同退火时间表作为变异算子，显示来自于 11 种 RNA 类别。总之，相比较遗传算法而言，退火算法能更好的收敛到全局最优，使 RNA 二级结构的预测准确度得到提高。但遗传算法的并行性和对邻域搜索的广泛性、灵活性也相当重要。所以，遗传算法和模拟退火算法相结合比单一使用其中任意一种方法预测 RNA 二级结构更具有优势。

随着机器学习算法的成熟发展，研究学者将 hopfiled 的神经网络算法应用与 RNA 二级结构的预测中。Liu Q 等人[20]提出了一种基于 HNN 的并行算法，将求解 RNA 二级结构等同于一种组合优化问题。利用 hopfiled 神经网络构造出能量函数，根据能量函数求解动力系统方程如式(1-1)所示，计算求解该方程的平衡点。通过参数调节，控制稳定茎区与非稳定茎区两者的比率。达到求解最优结构组合的目的。

$$E=\sum_{i=1}^{n}e_i v_i+\lambda\frac{\max|e_i|}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}v_i e_i \tag{1-1}$$

式中 i、j——碱基编号，$0\le i<j<n$，n 为 RNA 序列长度；

v_i——茎区 i 是否被选择，被选择为 1，反之为 0；

c_{ij}——两个茎区 i、j 是否相容，相容为 1，反之为 0；

e_i——茎区 i 的能量。

(2)基于能量次优的 RNA 二级结构预测算法。

随着对于 RNA 分子结构的深入，研究学者发现最小自由能模型预测的 RNA 二级结构并非真实结构。而真实的 RNA 二级结构存在于高于最小自由能能量 5~10%范围内次优结构集合中[21]，因此次优结构集合的研究算法对于 RNA 二级结构的预测尤为重要。

Ding 在 2003 年提出玻尔兹曼采样算法[22]。该算法是一种统计算法，通过构建 RNA 二级结构的玻尔兹曼集合，对 RNA 二级结构进行了严格而准确的采样。算法根据目前的热力学参数计算 RNA 二级结构的平均配分函数。利用递推采样过程中使用配分函数计算的条件概率，快速生成具有统计代表性的结构。该算法使用范围广效率高，通过抽样揭示了生物的结构特征。2004 年将这一算法搭载在 Sfold[23]平台上。

随着 RNA 二级结构采样算法的应用，研究者使用聚类算法对次优结构采样结果进行划分并获得代表簇类特征的特征结构，便于下一步研究进行。Ding 和 Lawrence 在 2005 年首次对次优结构进行聚类实验[24]，该算法首先计算这 1000 个结构之间的 BP 距离，构成距离矩阵。然后使用 Diana 层次聚类算法根据距离矩阵对数据进行划分，将相似结构划分为一簇。最后确定最小自由能结构为所在簇的特征结构。实验指出，最小自由能结构会出现在最大簇，次大簇，小簇甚至成为孤立点。因此，最小自由能结构不能成为代表一个簇的特征结构，所以在 2006 年 Ding 和 Lawrence 对该实验进行了修改，引入了质心结构作为簇的特征结构[25]，并应用于人类 mRNA 的聚类实验中。

Ding 和 Lawrence 的聚类算法，使用 BP 度量算法作为计算 RNA 二结构相似距离的方法，虽然简单易懂易于计算。但对比能力十分有限，只考虑了碱基对距离忽略了结构形状对于 RNA 二级结构的影响。所以 Phaedra 和 Zuker 等人提出了 RBP 评分算法[26]，并应用在 spectral k-means 聚类算法上。RBP 算法使用了一种相对放松的计算标准，只要碱基对在一定距离阈值范围内它们就是相似的。Zuker 的结果证明 RBP 比 BP 计算得出的簇更为稳定和有意义。文献[27]使用 RBP 算法计算 RNA 二级结构间距离应用于改进的 k-medoids 聚类算法中。该算法排除数据中的异常点，采用逐步扩大中心搜索范围的方法在替换聚类中心，提升了聚类的效果。文献[28]采用改进的 DBSCAN 聚类算法应用于基于 RBP 算法预测 RNA 二级结构，更好的处理 RNA 二级结构的密度数据集。

随着算法的不断改进，研究者发现较低粒度、较高抽象的二级结构表示对于 RNA 二级结构预测精度的提高有更大的贡献。表明随机误差和影响准确性的系统误差都可以通过二次结构的“模糊”的观点降低[29]。RNAshape[30]通过抽象出 RNA 二级结构的内环、凸包环以及茎区位置和长度来表示其拓扑结构和形状。将嵌套和邻接信息保留了下来表示在抽象形状中，如图 1-1 所示。该图为 RNAshpes 工具软件绘制[31]，

图的上半部分为 RNA 二级结构的多边形平面图，图的下半部分“[[ ] [ ]]”为抽象形状。其中，一对“[ ]”表示由碱基对 *i.j* 开始的封闭子结构。使用这种抽象形状表示结构，将具有相同形状的结构聚集在一起成为一簇。每个簇内结构的共有形状作为该簇的签名，包含结构的数量作为频率，shrep 作为代表性结构[31]，即类别内自由能最小结构。这种粗粒度的分析方式使研究人员能够获得其感兴趣的拓扑结构，对于序列特征的探索有重要意义。

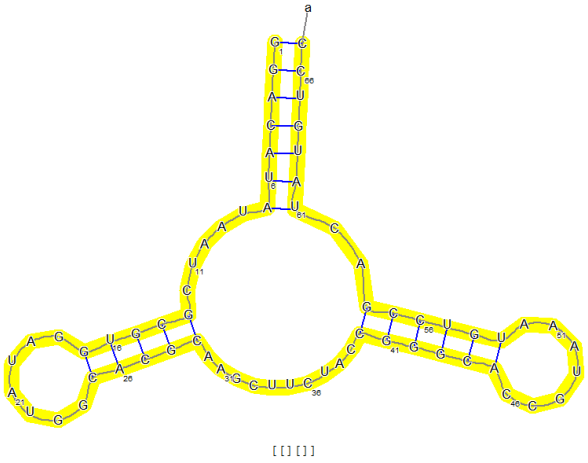


图 1-1 RNA 二级结构的多边形表示图和抽象结构表示

RNAHeliCes[32]是由 RNAshape 扩展而来，精度在 Sfold 与 RNAshape 之间。该算法的采样方式不同于玻尔兹曼算法随机的采样模式，而是确定性的列举了所有低能量结构。在形状的抽象上，RNAheliCes 在每个环结构框架中添加了一个索引值来区分不同环位于茎区的不同位置。索引值的数值是该环区的闭合碱基对所处位置的平均数，即设 *i.j* 为该闭合环区的闭合碱基对，则其索引值为 $(i+j)/2$。虽然相比较 RNAshape 算法 RNAHeliCes 算法更具有代表性意义，但次优空间集合的数量限制了其实际应用[29]。

1.2.2 基于多序列研究算法

基于多序列的研究算法以同源 RNA 的多个序列为输入，通过序列比对的方法，观察不同位置碱基的变化来确定同源序列的共有结构即共识结构，从而预测 RNA 二级结构的保守碱基对。按照 RNA 序列比对和预测的先后顺序可以分为先比对，后预测；先预测，后比对；边比对，边预测。

(1)先比对，后折叠。该方法假定 RNA 分子具有同源相似性，即功能相同的 RNA

序列具有相似结构。在RNA分子演化过程中，在配对碱基上可能会产生补偿性突变，则这对突变碱基称为协同变异。预测RNA二级结构时，需要找到协同变异的碱基对，并以其为中心，在附近区域寻找碱基配对可能性高的结构。通常这一过程需要构建一个数学模型支持，常见的模型有共变模型[33]和随机上下文无关语法模型[34]，这些模型可以从结构功能进化关系等方面反映结构的生物学特性。基于序列比较分析算法预测结果的精确度仅次于实验测定的RNA结构，但这种算法预测时需要已知一定数量的同源序列，难以正确预测先验知识较少的RNA序列。并且在处理序列联配问题上会产生较高的时间复杂度和空间复杂度，无法应用于长序列预测。

(2)先预测，后比对。该方法首先输入多个序列的最小自由能结构，然后通过搜索序列中共有的最低能量结构为共识结构，来实现最终预测。Tabei[35]等人提出基于多序列比对的非编码 RNA(ncRNAs)预测算法，该算法通过局部多重比对方式检测ncRNAs。Sven Siebert[36]在RNA预测软件MARNA中采取成对比对策略多序列比对提取共识结构的预测方法。这种先预测，后比对的预测顺序，其精确度依赖于比对算法的区分能力，而比对算法本身就是研究中的难题。

(3)边预测，边比对。这种方法将 Sankoff 比对算法和 Nussinov 预测算法结合，一边比对一组RNA序列，一边对该序列的二级结构进行预测，可以得到一个对比结构和一个共识结构。Mathews 等人提出的 Dyalingn 预测软件[37]实现了该算法在两个序列之间的比对，通过限制序列间的比对位置来降低算法的时间复杂度，因此该软件不能预测碱基个数较多的RNA分子。

1.3 本文的主要研究内容

本文通过对常用的RNA二级结构聚类算法进行分析，结合RNA二级结构集合数据特点，对比总结传统聚类算法存在的优缺点，为算法的设计提供参考。

首先，对RNA二级结构距离算法进行研究。由于RNA二级结构间距离是聚类划分的依据，RNA 二级结构距离计算算法能否有效识别 RNA 二级结构之间的差异直接影响了聚类的效果。之前的聚类算法采用BP算法或RBP算法计算RNA二级结构距离，仅考虑碱基对位置差异，忽略了形状的不同。因此，本文提出一种基于RNA二级结构形状-碱基对距离计算算法(RNA shape distance and base pair，Rsd-bp)，从RNA 二级结构形状和碱基对差异两个方面计算 RNA 二级结构间的距离，并对该算

法进行多进程优化，提升算法效率。

然后，对 RNA 二级结构聚类算法进行研究。由于 RNA 二级结构集合在聚类之前无法估计数据形状和是否存在孤立点即噪声点，传统聚类算法采用随机选择的方式初始化中心点，易陷入局部最优解。因此针对这一问题本文提出一种半监督聚类算法 SS-medoids^{+}算法，该算法结合半监督聚类的思想优化 k-medoids 聚类算法，根据监督信息进行中心点初始化和数据划分，并改进 k-medoids 算法中心点更新规则，缩小中心点更新时需要查找数据的范围，提高了聚类的效果和计算效率。

1.4 论文组织结构

论文组织架构具体如下。

第2章主要介绍相关知识。阐述了RNA概念；RNA二级结构的定义、组成、表示方法、传统距离计算方法；聚类算法、半监督聚类算法的基本概念。

第3章主要提出了一种计算RNA二级结构形状-碱基对距离的算法Rsd-bp算法。该算法从RNA二级结构形状差异和碱基对两个方面计算RNA二级结构间距离，能更好的区别RNA二级结构之间的差异。然后基于多进程计算对该算法进行优化，缩短计算时间，提升算法效率。

第4章主要提出一种半监督聚类算法SS-medoids+算法，该算法首先根据Rsd-bp算法的计算结果构建距离矩阵和约束集合，并对约束集合进行预处理，构造监督信息。然后根据监督信息进行中心点初始化和数据划分。最后改进k-medoids算法中心点更新规则，缩小中心点更新时需要查找数据的范围。

第5章即实验部分，结合实验验证本文提出算法。

最后结论。对本课题的研究内容和成果进行总结，提出文章中存在的问题和可以改进的地方。

第 2 章 相关知识

2.1 RNA 基本概念

2.1.1 RNA

RNA(核糖核酸)是由核糖核苷酸经磷酯键缩合而成长链状分子，与脂质、蛋白质和碳水化合物一起构成已知生命形式所必需的四种主要大分子。如图 2-1 所示 RNA 分子由一个核糖核苷酸分子、磷酸、核糖、碱基构成，RNA 碱基主要有 4 种，即 A 腺嘌呤、G 鸟嘌呤、C 胞嘧啶、U 尿嘧啶。根据碱基配对原则，RNA 自身折叠形成空间结构来行使生物学功能。RNA 碱基配对规则为 A-U、C-G、G-U，其中 G-U 碱基对通常不稳定，又被称为摆动基对。

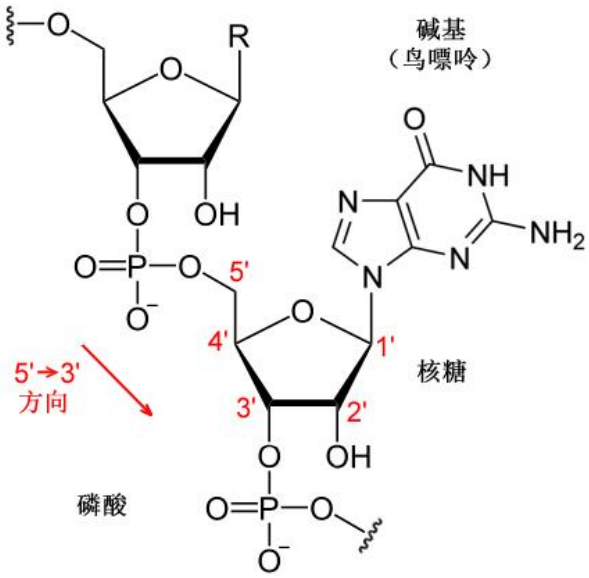


图 2-1 RNA 化学结构

根据功能不同，RNA 可以分为将蛋白质信息从 DNA 转录到核糖体的信使 RNA(mRNA)、用于翻译 mRNA 所携带的蛋白质序列信息的核糖体 RNA(rRNA)、携带并转运氨基酸功能的转录 RNA(tRNA)、不参与蛋白质编码的非编码 RNA(ncRNA)。

2.1.2 RNA 二级结构

RNA 的二级结构通常可分解成茎和环。RNA 碱基互补配对堆积形成的区域成为茎区，茎区与茎区之间未配对的单链称为环结构。环结构与茎区交替出现构成了 RNA

二级结构。环结构一般分为发卡环(hair-loop)，凸包环(bulge-loop)，内环(inner-loop)和多分支环(mult-loop)。如图 2-2 所示。

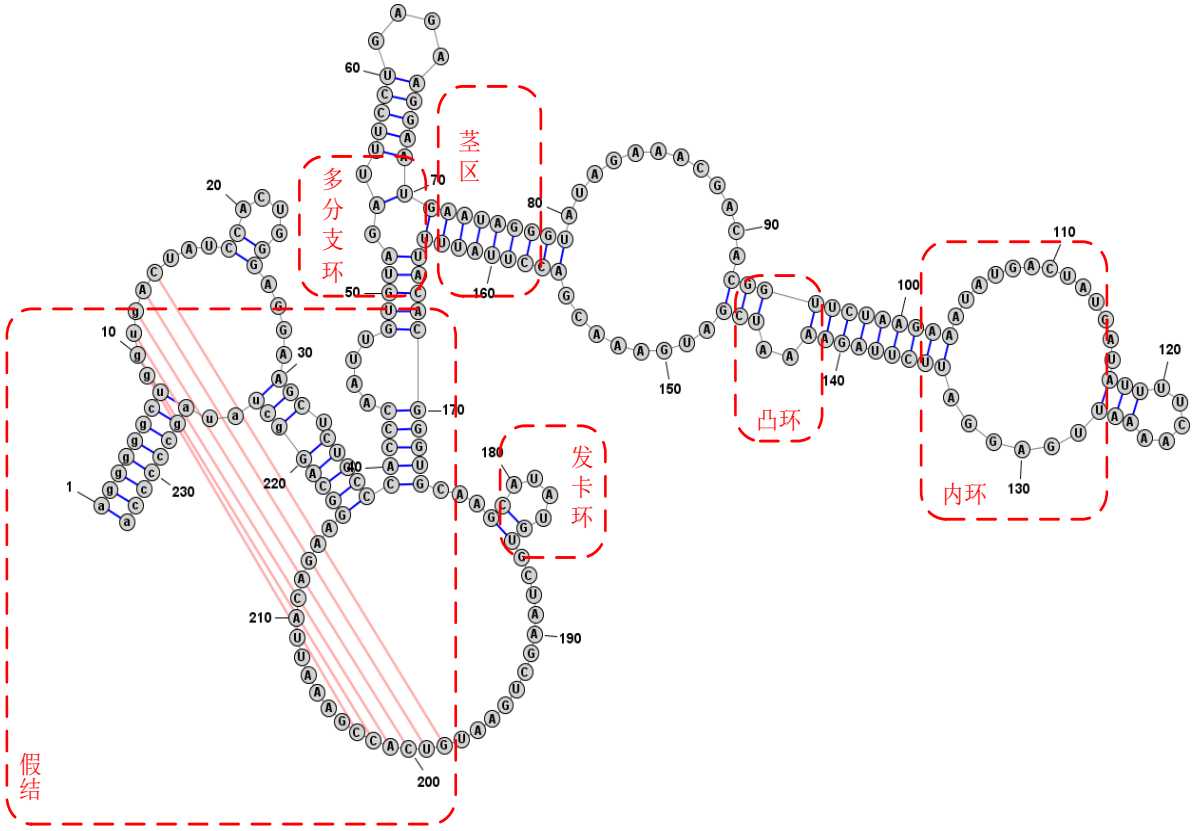


图 2-2 Methanococcus marapaludis RNA 的二级结构示意图

设 RNA 序列为 N，$N=\{1,2,3,\ldots,i,\ldots n\}$，1 到 n 分别表示 RNA 碱基的序号，1 为 5’端，n 为 3’端。将该序列形成的二级结构定义为碱基对$(i.j)$的集合 S 如式 2-1 所示。

$$S=\{(i.j)\,|\,1\le i\le j\le n\} \tag{2-1}$$

为确保其形成的二级结构可行有效，应满足以下约束[38]。

(1)一个碱基只能与一个碱基配对；

(2)两个配对的碱基之间必须间隔三个碱基；

(3)若不包含假结，则不可出现碱基对$(i.j)$和$(k.l)$，其中$i<k<j<l$；

(4)如果违反(3)会形成假结，则定义其为三级结构基序。

为了表示 RNA 二级结构，将二级结构特征抽象成图形。如图 2-3 所示，常见二级结构图形表示方法[38]有多边形图、圆顶图、圆圈图、点阵图、山峰图、点括号图和树形表示法。

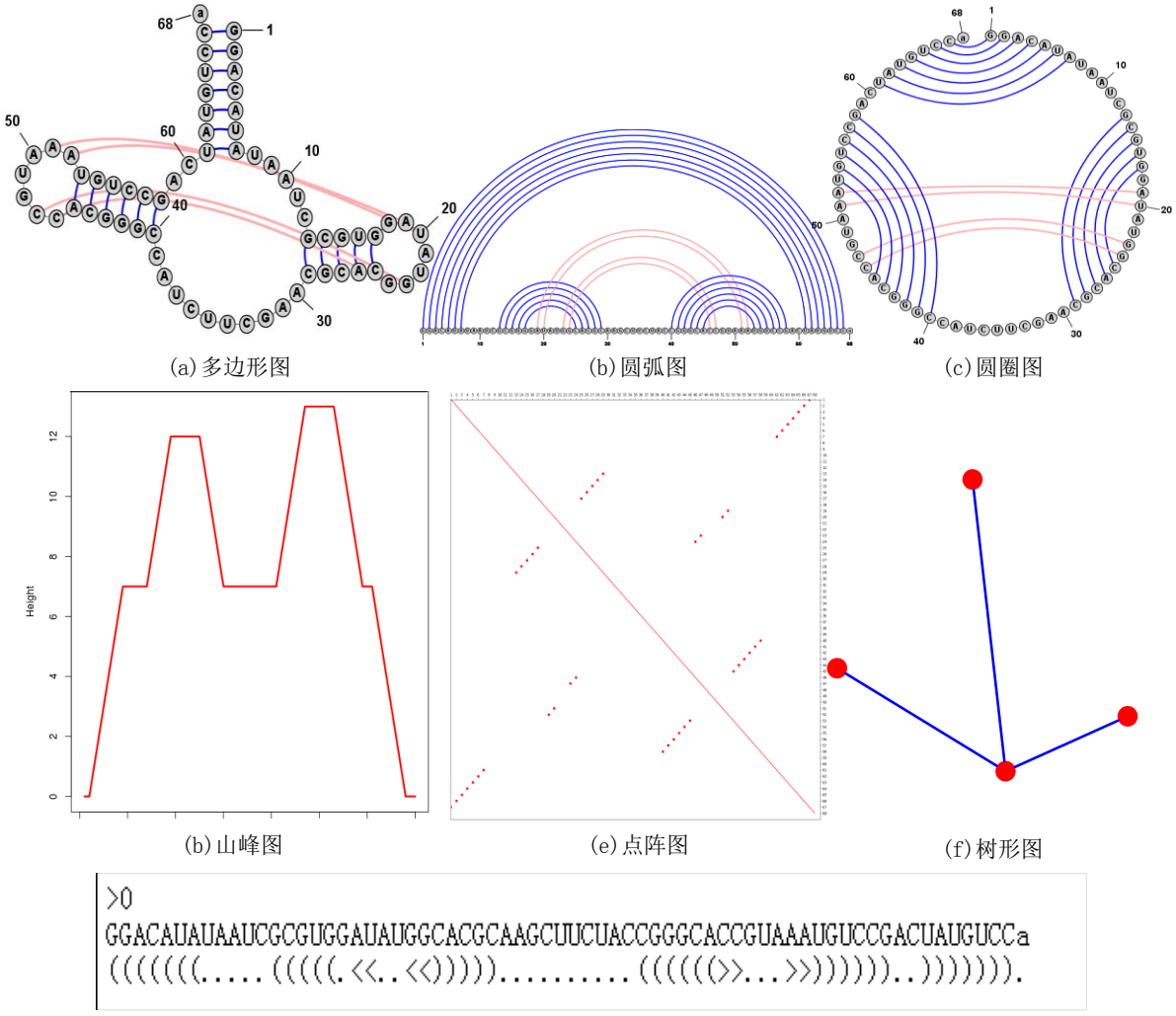


图 2-3 RNA 二级结构常见表示方法

多边形图(见图 2-3(a))是最为直观和经典图形表示的表示方法，其中用顶点代表单个碱基，用小短线将顶点依次相连表示序列，用垂直于序列方向线段连接碱基表示配对碱基。多边形图可以清楚的表示茎环结构的关系。但对于较大的二级结构绘制多边形图往往会使布局变得混乱，如何构造良好的布局成为重要问题。

圆弧图(见图 2-3(b))将 RNA 序列表示为一条直线，点表示碱基，使用圆弧线将配对碱相连，若圆弧出现交叉则说明二级结构中存在假结。圆弧图常用于表示输出或分析含有假结的二级结构。

圆圈图(见图 2-3(c))与圆弧图类似，用圆圈表示 RNA 序列，用圆弧将配对碱基相连，若含假结则图中圆弧相交。圆圈图常用于结构的输出。

点阵图(见图 2-3(d))即螺旋区点阵图，把 RNA 序列同时作为横纵坐标，通过查

找标记配对碱基用点表示，将点用线串连，这些线与横纵坐标呈 45°角的斜线的碱基对为茎区。点阵法常应用于 RNA 结构的比较分析。

山峰图(见图 2-3(e))，用高度和位置关系表示二级结构。其中对称的斜坡表示茎区，峰顶表示发卡环的数量，山谷表示环区或单链碱基。

树形图，根据表示二级结构的详细程度不同可以有多种表示方式。如图 2-3(f)所示提供了实际二级结构外观的简化图，将 RNA 二级结构的起始端结束端(5’端和 3’端)和环区简化为圆点，茎区简化为线，点线相连形成一棵树。树形图常与图论方法结合用于计算 RNA 二级结构间距离。

点括号图(见图 2-3(g))，最常见的表示方法是用由括号和点组成的长度为 n 的序列表示长度为 n 二级结构。在第 i 位置处用“(”，第 j 位置处用“)”表示配对碱基 i.j，未配对的碱基表示为点。点括号图常与山峰图用作 RNA 结构的计算和分析的等价图。

除了这些图形表示方式，在 RNA 二级结构的研究算法中常以 CT 文件存储 RNA 二级结构，常见 CT 文件包含 6 列数据：第 1 列为索引列，对碱基的逐个编号；第 2 列表示序列从 5'端到 3'端的碱基信息；第 3 列是该碱基的前一个碱基的编号；第 4 列是该碱基的后一个碱基的编号；第 5 列表示该碱基是否而已与其他碱基配对，“0”表示不能够与其他碱基配对，非“0”表示与该碱基配对的碱基的编号；第 6 列与第一列相同。

2.1.3 RNA 二级结构常见相似度距离计算算法

(1)BP 距离

Ding 提出的 RNA 二级结构层次聚类算法[24,25]中使用 BP 距离作为衡量 RNA 二级结构间差异的依据，BP 距离越大说明两个结构之间的差异越大。该算法计算简单易于理解，计算公式如式(2-1)所示。

$$\begin{cases} D_{bp} = I_1 / I_2 \cup (I_2 / I_1) \\ D_{bp} = \sum\limits_{1 \le i < j \le n} S_{ij}^1 - S_{ij}^1 \end{cases} \tag{2-1}$$

式中 S^1、S^2——编号为 1、2 的二级结构；

S_{ij}^1、S_{ij}^2——二级结构的当前碱基对；

I_1、I_2——二级结构的最大碱基对集合。

(2)RBP 评分

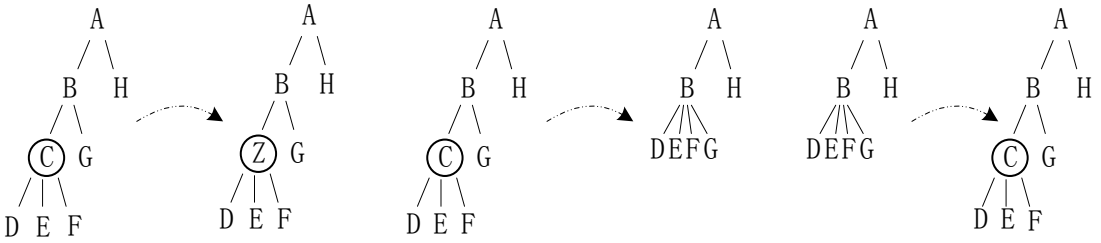
为弥补 BP 距离对 RNA 二级结构间差异区分能力的不足，Zuker 等人提出了 RBP 算法[26]。其计算方法如下。

设长度为 n 的 RNA 单链碱基序列为 L，L={$r_1,r_2,r_3...,r_n$}，当 r_i 与 r_j 配对时表示为 $i.j$，$1 \le i < j \le n$。则 $i.j$ 到 $i'.j'$的距离为 $d(i.j,i'.j') = \max\{|i-i'|,|j-j'|\}$。$i.j$ 到二级结构 S 之间距离为 $\delta(i.j,S) = \min d(i.j,i'.j')$。则二级结构 S_A 和 S_B 的 RBP 距离，即 $\rho(S_A,S_B) = \min\{m \in Z \mid \Delta_k \le tm \text{ if } k \ge m\}$，公式中的 m 为长度为 k=|S_A|+|S_B|递减有序的非负整数数组，数组的值为 $\{\delta i.j,\ S_B, i.j \in S_A\}$ 和 $\{\delta i.j, S_A, i.j \in S_B\}$。通过对数组 m 中的 k 个数值按递减顺序进行排序，计算求的唯一定义的数组，得 $\Delta(S_A,S_B) = (\Delta_1,\Delta_2,\ldots,\Delta_k)$，根据对称性有 $\Delta(S_A,S_B) = \Delta(S_B,S_A)$，$\Delta(S,S) = (0,0,\ldots,0)$。公式中的 t 为设定的一个非负实数，即松弛系数。当 t=0 时 $\rho(S_B,S_B) = D_{bp}(S_B,S_B)$，不同的 t 值对应不同的结果，可以通过改变 t 值可以使计算结果更加灵活，便于分析。

RBP 算法虽然在一定程度上弥补了 BP 算法的缺陷但具有极高的时间复杂度，在处理大容量 RNA 二级结构计算时十分耗时，而且仅凭借松弛系数放宽 RNA 相似度距离计算的要求缺乏客观依据，在 RNA 二级结构相似度距离的计算中易产生误差。

(3)树编辑距离

RNA 二级结构的树编辑距离[39]是指将 RNA 二级结构抽象为树模型表示，通过一系列的编辑操作将一棵树输入树转化为另一颗输入树的最低总成本。一般编辑操作有替换、插入、删除如图 2-4 所示。



(a)替换　　(b)删除　　(c)插入

图 2-4 树编辑操作

每个编辑操作都会与一个 cost 相关联，cost 的值可根据算法需求自定义设置，当一棵树被编辑到另一棵树时，编辑操作成本的累积总和构成编辑总成本。分配成本时，需尽量减少成本以寻求最小结构距离。编辑距离常用于计算同源的不同序列 RNA 折叠形成的 RNA 二级结构之间的距离，来发现同源 RNA 的相似结构。

2.2 聚类

2.2.1 聚类概念

聚类分析实质上是对一组对象进行分组，使得同一簇中的对象在某种意义上相对于其他簇中的对象更为相似[44]，即将足够相似的对象分配为一簇，并将不同簇加以明确的区别和分隔。聚类可以表述为多目标优化问题，根据目标任务的特点，选定合适的聚类算法和参数设置，通过实验失败或交互式多目标优化的过程迭代，修改数据预处理和模型参数，直到结果达到所需属性。

将聚类问题一般表述如下。

设大小为 n 的样本集合为θ，$\theta=\{x_1,x_2,\ldots,x_n\}$，其中对于每个样本 x_i 是一个含有 n 维特征向量无标记样本，$x_i=(x_{i1},x_{i2},\ldots,x_{in})$。聚类算法通过对样本间的空间位置关系进行距离计算，划分样本集合θ成 k 个互不相交簇C_l，$l=1,2,3\ldots,k$。其中，即$C_l\cap C_{l'}=\varnothing$，$l\neq l'$；$\theta=\bigcup_{i=1}^{k}C_l$，$C_l=\{x_{l1},x_{l2},\ldots,x_{li}\}$。每个 x_{li} 都标记为同一标签，不同簇为不同标签。若 $x_{il}\in C_l$，则 $x_{il}\notin C_k$，$k\neq l$。

2.2.2 常见聚类分类

常见的聚类算法有层次聚类、原型聚类、密度聚类等。

(1)层次聚类算法，此类算法通过计算节点间距离并按由低到高排序，逐步连接的方式进行数据划分。根据距离的层次分解的不同又可以分为自顶而下的分裂层次聚类和自底而上的凝层聚类[45]。分裂算法初始化时所有数据看成一个簇，然后通过计算数据间的距离找出簇中最远距离，划分成两个簇，然后不断重复这一过程直到达到预期目的。而凝层聚类与之相反，该算法初始化时将每一个数据看做一个簇，找出最小距离进行合并，并重复至达到预期条件停止。文献[24]使用了 DIANA 分裂层次聚类算法对 RNA 二级结构进行了聚类。

(2)原型聚类算法，此类算法通常使用一组原型刻画数据关系，是十分常见的聚类算法。一般情况下，算法首先对原型进行初始化，然后根据原型进行迭代更新，求解数据划分至最优原型。常见的原型聚类算法有基于中心划分的 k-means 算法、k-medoids 算法，基于概率模型的的高斯混合聚类算法[47]等。其中 k-means 算法又称为 K 均值聚类算法[42,43]，该算法可将数据划分为 k 个组，聚类质心为数据的均值中

心。而 k-medoids 算法[46]则限制质心为数据集成员，相比于 k-means 算法具有更强的鲁棒性。基于中心划分的原型聚类算法通常需要事先设定聚类簇数 k，而 k 值的设定会影响聚类结果的准确性，这是该算法的最大缺陷。

(3)基于密度的聚类[48]，是基于数据密度进行聚类划分。该算法假定类别可以通过样本分布的紧密程度决定，同一类别内样本紧密相连。这类算法能抵抗稀疏区域产生的噪声干扰，适合任意形状的数据集合。

2.3 半监督聚类

聚类算法通常是无监督算法，聚类过程中只能访问描述每个数据的一组特征，没有给出关于数据划分的任何信息。然而，在实际应用领域中，实验者拥有一些可用于聚类数据的先验知识，传统无监督聚类算法无法利用这些信息。因此，将先验知识集成到聚类算法中可以提高聚类效果。

在半监督聚类[49]中根据引入先验知识的不同形式分为两种类型，即基于标记数据的半监督聚类和基于约束的半监督聚类。

(1)基于标记数据的半监督聚类[50]数据集除了大量的无标记训练样本外，还包含一些少量有标记的样本。在聚类过程中将这些有标记样本作为“种子”用于初始化聚类中心点，并通过不断迭代更新改变这些种子的簇隶属关系，以达到帮助聚类中心初始化，加快聚类收敛的效果，从而更快的得到稳定的聚类结果。

(2)基于约束的半监督聚类[50,51]，通常定义两种类型的约束集合，在聚类过程中通过满足两类集合约束控制数据划分过程。这种方式可以使聚类结果对数据产生合适的划分，满足约束信息的特定需求。

在基于约束的半监督聚类中通常定义两种类型的约束集合[52,53]，即必连集合，勿连集合。根据数据约束的特性有如下定义。

设样本集合为 D，集合内含有 n 个样本x_i，则 $D=\{x_1,x_2,..,x_i,\ldots x_n\}$ 通过聚类划分形成的类别名称为C_i，i 为类别编号。

定义 2-1：设数据集合中有若干对数据对 (x_i,x_j)，已知 (x_i,x_j) 必不存在于聚类的同一簇中，则称这若干对约束数据对的集合为勿连集合(cannot-link,CL)，约束数据间存在正关联约束，如公式(2-2)所示。

$$CL=\{\ldots(x_i,x_j)\ldots\} \tag{2-2}$$

定义 2-2：设数据集合中有若干对数据对(x_h, x_k)，已知(x_h, x_k)必存在于聚类的同一簇中，则称这若干对约束数据对的集合为必连集合(must-link,ML)，约束数据间存在负关联约束，如公式(2-3)所示。

$$ML = \{\ldots(x_h, x_k)\ldots\} \tag{2-3}$$

由于数据之间的关系是互相对称存在的，若x_i属于类别C_i，x_j属于类别C_j，则有如下性质[51,52]。

(1)对称性，如公式(2-4)所示。

$$\begin{aligned}(x_i, x_j) \in ML &\Rightarrow (x_j, x_i) \in ML \\ (x_i, x_j) \in CL &\Rightarrow (x_j, x_i) \in CL\end{aligned} \tag{2-4}$$

(2)传递性，如公式(2-5)所示。

$$\begin{aligned}(x_i, x_j) \in ML \,\&\&\, (x_j, x_k) \in ML &\Rightarrow (x_i, x_k) \in ML \\ (x_i, x_j) \in ML \,\&\&\, (x_j, x_k) \in CL &\Rightarrow (x_i, x_k) \in CL\end{aligned} \tag{2-5}$$

半监督聚类通常是在无监督聚类模型的基础上直接引入这些约束信息，进行强制性约束，如约束 k 均值算法[52]，在对数据点划分过程中，优先检验数据点是否存在约束关系，将数据点划分到满足约束关系中的簇内。虽然使用这些约束可以引导聚类算法将数据划分到更适合的簇中，但如何选取约束进行约束聚类仍为一个难题，文献[54]证明严格的遵从约束划分并不一定会产生好的聚类效果。因为在选定的约束集合中若存在错误约束会引起聚类失败。所以如何筛选出约束集合内的错误信息十分重要。

本文对原有半监督聚类划分方式进行改进，分析约束集合导致聚类失败的因素，根据 RNA 二级结构集合特点，预先处理约束集合生成合适的约束信息，在强制数据成对约束的同时，保证聚类的稳定性和结果的准确性。

2.4 本章小结

本章首先介绍了 RNA 的概念知识，包括 RNA 构成、RNA 二级结构相关概念。详细的解释了 RNA 二级结构的组成、表示方法和距离计算方法为第 3 章 RNA 二级结构距离计算算法的研究提供理论知识和铺垫。然后对聚类算法和半监督学习的相关理论进行介绍为第 4 章提出 RNA 二级结构的半监督聚算法做铺垫。

第 3 章 基于 Rsd-bp 距离的多个 RNA 二级结构距离计算

3.1 问题分析

RNA 二级结构间距离是聚类划分的重要依据，有效的 RNA 二级结构距离计算算法可以使聚类划分效果得到提升，令同簇内 RNA 二级结构更加相似，异簇内结构差异更大。

Ding 提出的聚类算法使用 BP 距离计算 RNA 二级结构间距离，BP 距离越小，说明 RNA 二级结构越相似。该算法计算方式虽然简单易懂，但仅考虑碱基对差异，忽略了二级结构间的形状差异。Zuker 提出使用 RBP 算法，在一定程度上弥补了 BP 算法缺陷，但该算法具有很高的时间复杂度，对于长序列大容量的 RNA 二级结构集合间距离的计算十分耗时。

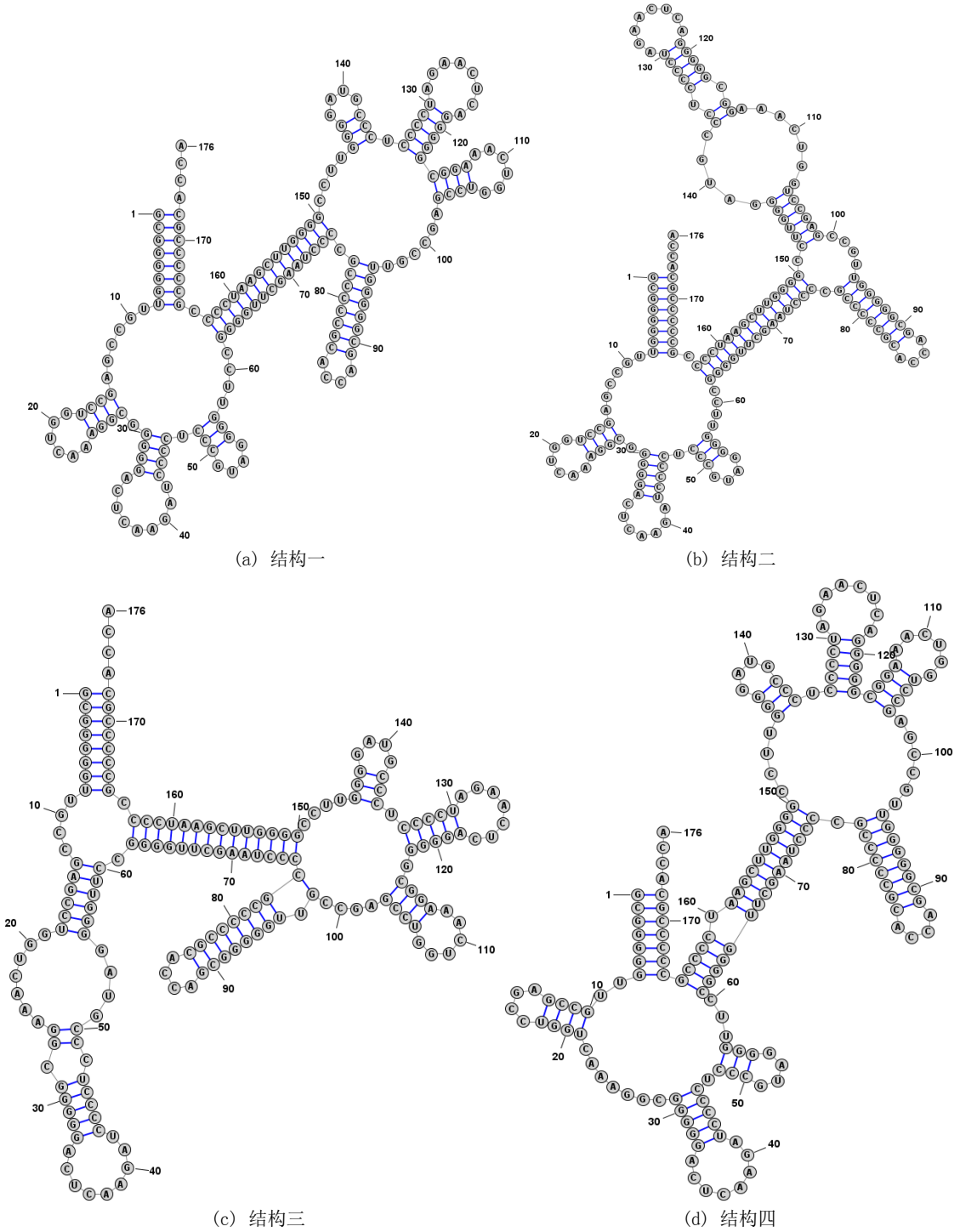
以来自于 RCSB 蛋白质数据库的 RNA 序列(THE CRYSTAL STRUCTURE OF LEUCYL-TRNA SYNTHETASE AND TRNA COMPLEX，简称 CRYSTAL 序列)为例，如图 3-1 所示，图中所有二级结构由 sfold2.0 根据 CRYSTA 序列随机生成，并由 RNAStructure 软件绘制的 RNA 二级结构图。

图 3-1(a)与图 3-1(c)之间的 BP 距离为 27，图 3-1(a)与图 3-1(d)之间的 BP 距离也为 27，但从形状结构组成上看图 3-1(a)与图 3-1(d)之间更为相似。而图 3-1(b)和图 3-1(c)、图 3-1(a)从结构上差异度似乎相同，但图 3-1(a)与图 3-1(b)的 BP 距离为 16，从碱基对的差异程度上看似乎图 3-1(a)与图 3-1(b)更为相似。综上，仅从 RNA 二级结构的形状或碱基对其中一个方面计算二级结构距离，会产生一定的误差。

因此，提出一种基于形状-碱基对(RNA shape and distance-base pair,Rsd-bp)的比对算法，该算法从形状和碱基对这两方面来计算 RNA 二级结构间距离，同时适用于多个结构之间的计算，可以体现二级结构在空间集合中的特征。

3.2 Rsd 算法

Rsd 算法是基于树模型的 RNA 二级结构形状距离计算算法，该算法首先将两个不同的 RNA 二级结构抽象为两棵有序树，然后计算这两棵有序树转化为同一棵有序树的编辑距离，该编辑距离为两个 RNA 二级结构间的距离，即 Rsd 距离。Rsd 距离越小，表示 RNA 二级结构越相似。



(a) 结构一　(b) 结构二

(c) 结构三　(d) 结构四

图 3-1 相同序列折叠的 RNA 二级结构

3.2.1 Rs 树模型构建

Rs 树是一种基于 RNA 二级结构形状抽象的树模型。在不考虑碱基对因素下，忽略茎区长度和环的大小，将多分支环，发卡环，内环和凸起抽象为符号，突出表

现结构中茎环位置关系。根据 RNA 二级结构的基本信息有如下定义。

定义 3-1：封闭结构 若碱基 i、j 互补配对，$0 \le i < j \le n$，i+1 或 j-1 为不配对的自由碱基，则将 $i.j$ 称为由 i 到 j 的子结构的闭合碱基对，i 到 j 之间的子结构称为封闭结构。

定义 3-2：分支结构 若在以闭合碱基对 $i.j$ 为起点的封闭结构中，若存在碱基对 $h.k$、k+n 为闭合碱基对 n>0，则称这些闭合碱基对为起点的封闭结构为分支结构。与闭合碱基对 $i.j$ 距离最近的两个或多个闭合碱基对引起的封闭结构为封闭结构 $i.j$ 的最大分支结构，如图 3-2 所示。

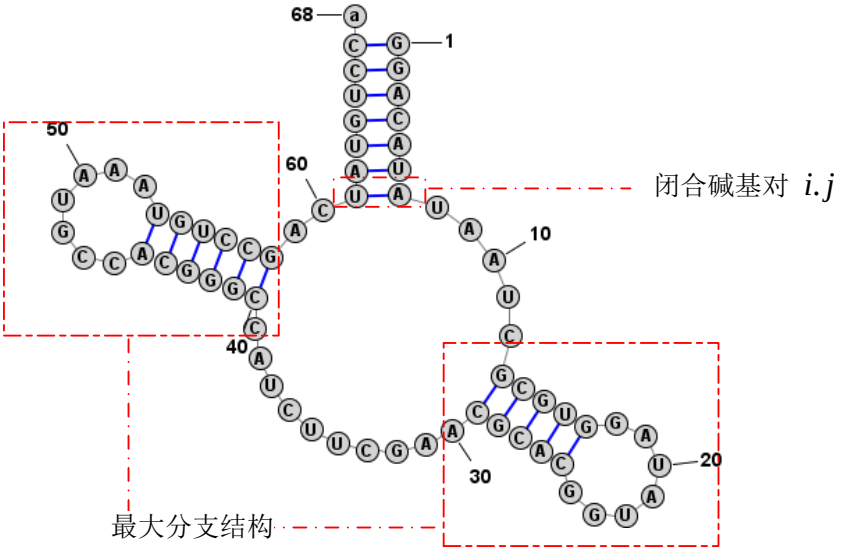


图 3-2 最大分支结构示意图

RNA 二级结构信息抽象为 Rs 树模型时，按先序遍历方法，顺序编码生成有序结构树。定义根节点为起始封闭结构为 R_1，即靠近 3' 到 5' 端分第一对闭合碱基为起点的封闭结构。定义子节点分别为父节点的最大分支结构，当节点为根节点时，起始封闭结构之前存在左侧凸包环，右侧凸包环，内环时，将这些环结构抽象为符号，依次存储在根节点上。当子节点与父节点之间存在这些环结构时，将这些环结构抽象为符号，依次存储在子节点上。父节点与子节点的关系如公式(3-1)所示。

$$R_i = \{\{\Phi_m + R_m\} \mid 0 \le i < m \le |R|\} \tag{3-1}$$

式中 R_i、R_m ——当前指向的节点、当前节点的子节点，i=1 时为根节点，当 R_i 为叶子节点时，若不存在 R_m，$R_i = \{\Phi_m\}$；

Φ_m ——节点到其父节点之间的抽象子结构的集合，Φ_m 中可能包含 BL(左侧凸环)、BR(右侧凸环)、IL(内环)。

设长度为 n 的 RNA 二级结构的序列 L，$L=\{i,1\le i\le n\}$，i 为碱基编号，序列的开始端 5'指向 1，结束端 3'指向 *n*。*i.j* 表示配对碱基，则 BL、BR、IL 的判断方法如公式(3-2)所示。

$$\begin{cases} BL=\{i.j,i'.j'\mid i+m=i',j-1=j',m\ge 1,1\le i\le i'\le j\le j'\le n\} \\ BR=\{i.j,i'.j'\mid i+1=i',j-m=j',m\le 1,1\le i\le i'\le j\le j'\le n\} \\ IL=\{i.j,i'.j'\mid i+m=i',j-m=j',m\ge 1,1\le i\le i'\le j\le j'\le n\} \end{cases} \tag{3-2}$$

式中 *i*、*j*、*m*——碱基编号；

i.j——碱基 *i*、*j* 配对。

如图 3-3 所示为 RNA 二级结构与 Rs 树的映射关系图。

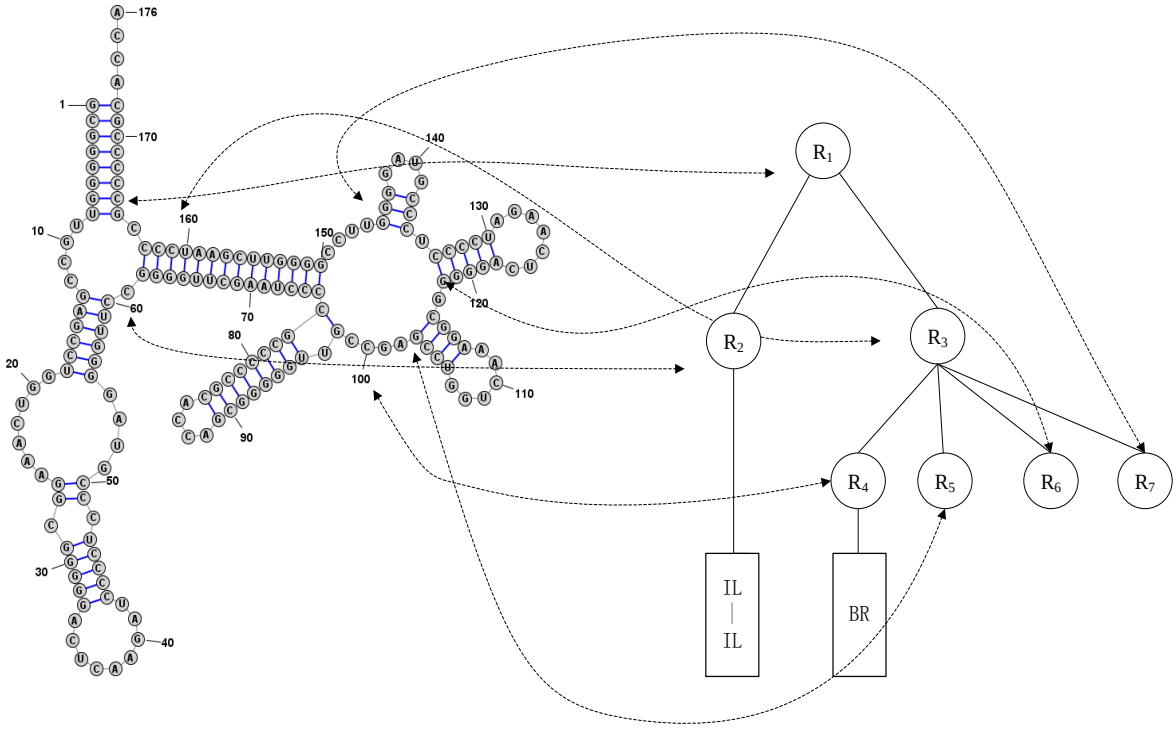


图 3-3 RNA 二级结构与 RS 树的映射关系图

根据映射关系，输入去除重复碱基对的 RNA 二级结构的碱基对列表，根据先序遍历算法生成 Rs 树模型，Rs 生成算法算法描述如算法 3.1 所示。

算法 3.1：Rs 树模型生成算法

输入：按 5’到 1’增序排序并去除重复碱基对的 RNA 二级结构碱基对列表 S。

输出：Rs 树模型 R。

BEGIN

1: function rstree(S,k)

```
2:      j←S[k][1],k←k+1,R←[ ]
3:      while s[k][0]<j and k<length(S)-1
4:        if s[k][0]<j and k<length(s)-1:
5:          k←k+1
6:        else if judge(s[k][0],s[k][1],s[k - 1][0],s[k - 1][1]) is BL
              #判断其是否为 BL 结构若是将其存储在上一节点
7:            R[-1] ← BL,k←k+1
8:        else if    judge(s[k][0],s[k][1],s[k - 1][0],s[k - 1][1]) is BR
9:            R[-1] ←BR,k←k+1
10:       else if  judge(s[k][0],s[k][1],s[k - 1][0],s[k - 1][1])  is branch then
11:           r,t←rstree(S[k:],0)          #递归函数 rstree，插入节点
12:           k←t+k
13:           R.append(r)            #将 r 添加到 R 中
14:       else is IL then
15:           R[-1] ←IL
18: return  R,k
END
```

算法 3.1 将去除重复碱基对的 RNA 二级结构碱基对列表 S 转化为带符号的抽象 Rs 树型结构存储。利用递归的方式构建树模型，其算法时间复杂度为 O($n$log$n$)，n 为碱基对对数。以树的形式存储二级结构，其空间复杂度为 O(r)，r 为生成的最大节点数。

3.2.2 Rsd 算法设计

Rsd 算法采用树编辑算法思想计算 RNA 二级结构距离，首先将两个 RNA 二级结构 S_i、S_j 抽象为 Rs 树 A，B，然后对这两棵树进行插入、转化操作，使这两棵树变成相同的树。例如，若在某一位置 A 中不存在节点 r，B 中存在，则在 B 这一位置插入节点 r，反之若 B 中不存在节点 r，A 中存在节点 r，则在 B 这一位置插入节点 r。若在某一位置 R_i、R_j 下都存在节点，但 A 这一位置为节点 r，B 这一位置为节点 p，则将 A 中的节点 r 转化成节点 p，B 保持不变。称这些插入、转化操作为编辑操作。最后根据编辑开销表(如表 3-1 所示)计算转化过程中的编辑开销总和，由树 R_i 到 R_j

的编辑开销总和即为 S_i、S_j 之间的 Rsd 距离。则 Rsd 距离的计算公式如公式(3-3)所示。

$$Rsd(A \to B) = \sum_{i=1}^{n} Rsd(A_i \to B_i) \tag{3-3}$$

式中 A、B——同序列生成的不同 RNA 二级结构的 Rs 树；

A_i、B_i——A、B 的第 i 节点上的字节点；

$Rsd(A_i \to B_i)$——A_i、B_i 之间的编辑距离。

Rsd 算法对 Rs 树模型 A，B 按先根顺序遍历节点，按 A 树中节点位置与 B 树对应位置进行编辑操作。由于 Rs 树的节点表示抽象环结构符号和子节点集合，所以 Rsd 算法的编辑操作分为对抽象结构符号集合的的操作和对节点集合的操作两类。

对于抽象结构符号操作如下。

(1) 若 Φ_{ai} 存在、Φ_{bi} 不存在，则插入 Φ_{ai} 到 Bi 中，记录开销；

(2) 若 Φ_{ai} 不存在、Φ_{bi} 存在，则插入 Φ_{bi} 到 Ai 中，记录开销；

(3) 若 Φ_{ai}、Φ_{bi} 同时存在且 $\Phi_{ai} \neq \Phi_{bi}$，将 Φ_{ai} 转化为 Φ_{bi}，记录开销；

(4) 若 Φ_{ai}、Φ_{bi} 同时存在且 $\Phi_{ai} = \Phi_{bi}$，则不对树 A 进行操作，记录开销为 0。

表 3-1 编辑开销表

|Distance|	BR	BL	IL	A_i	NULL
R	0	2	1	0	1
BL	2	0	1	0	1
IL	1	1	0	0	2
B_i	0	0	0	$\mathrm{Rsd}(A_i \to B_i)+1$	$\sum_{i}^{n} Rsd(B_i \to A_i)+1$
NULL	1	1	2	$\sum_{i}^{n} Rsd(A_i \to B_i)+1$	0

对节点的操作如下。

(1) 若当前位置下 A 存在节点 A_i，B 不存在 B_i 时，查询 A_i 节点下的多有节点，并插入到节点 B_{i-1} 位置后，记录开销；

(2) 若当前位置下 A 不存在节点 A_i，B 存在 B_i 时，查询 B_i 下所有节点依次插入 A 的 A_{i-1} 位置后。

由于 Rsd 算法只是模糊的计算两个结构间形状的距离，忽略环上自由碱基的个

数，仅对单链碱基形成的链数计算，所以定义这种编辑操作的开销单位为步长，1 步长等于删除或增加一侧单链。例如，一个内环有两条单链，而一个凸起环一条单链，所以由一个内环转化为一个凸起环步长为 1，详细的编辑开销如表 3-1 所示。

特别注意的是，在 Rsd 算法中在计算 *A* 到 *B* 的编辑距离时，将 *A*、*B* 共同编辑转化为相同的树。而树编辑操作的插入、转化操作均为互逆过程，所用开销相等，也就是说由树 *A* 转化为树 *B* 的编辑距离等于树 *B* 到树 *A* 的编辑距离。所以 Rsd(*A*→*B*)=Rsd(*B*→*A*)。综上，Rsd 的算法描述如算法 3.2 所示。

算法 3.2: Rsd 距离计算算法

输入：Rs 树 *A*、*B*，编辑开销 *cost*。

输出：*A*、*B* 间的 *Rsd* 距离。

```
BEGIN
1:  r1←A,r2←B
2:  function rsd(r1,r2)
3:      Rsd←0,i←0
4:      if   length(r1)>length(r2)
5:          n←length(r1)
6:      else   n←length(r2)
7:      while   i<n
8:          if   r1[i]or r2[i] is None   #A 或 B 的当前节点是否为空。
9:              if   judge r1[i] is None and r2[i] is banches
                    #B 的当前节点为空且 A 的当前节点有子节点。
10:                 build r←[ ],insert r in r1[i],Rsd←Rsd+function rsd(r1[i],r2[i]),i←i+1
                    #创建新的节点 r 并插入 A 的当前位置，递归函数 rsd(r1[i],r2[i])
11:             else if   judge r1[i] is None and r2[i] is abstract loop
                    #A 为空且 B 为抽象环结构
12:                 instert r2[i] in r1[i],Rsd←Rsd+cost,i←i+1
13:             else if   r1[i] is banches r2[i] is None
14:                 build r←[ ],instert r1[i] in r2[i],
                    Rsd←Rsd+function rsd(r1[i],r2[i]),i←i+1
```

15: else if judge *r2*[*i*] is None and *r1*[*i*] is abstract loop
#*B* 的当前节点为空且 *A* 的当前节为抽象子结构。
16: instert *r1*[*i*] in *r2*[*i*]
17: *Rsd*← *Rsd*+*cost*,*i*←*i*+1
18: else *i*←*i*+1
19: else
20: if *r1*[*i*] and *r2*[*i*] are abstract loops ,and *r1*[*i*] is not *r2*[*i*]
21: *r1*[*i*]←*r2*,*Rsd*←*Rsd*+*cost*,*i*←*i*+1
22: else if *r1*[*i*] ,*r2*[*i*] are banches and *r1*[*i*] is not *r2*[*i*]
23: *Rsd*←*Rsd* + rsd(*r1*[*i*], *r2*[*i*], *i*←*i*+1),*i*←*i*+1
24: else if *r1*[*i*] is banches *r2*[*i*] is abstract loops
25: insert *r2*[*i*] in *r1*[*i*],*Rsd*←*Rsd*+*cost*,*i*←*i*+1
26: else if *r2*[*i*] is banches *r1*[*i*] is abstract loops
27: insert *r1*[*i*] in *r2*[*i*],*Rsd*←*Rsd*+*cost*,*i*←*i*+1
28: else *i*←*i*+*1*
29: return *Rsd*
END

算法 3.2 将两棵不同的 Rs 树 A、B 转化为相同的一颗 Rs 树，通过计算两棵 Rs 树转化所需的编辑开销之和，作为 A、B 表示的 RNA 二级结构的 Rsd 距离，即结构形状之间的形状距离。当 Rsd 距离越小时，两个 RNA 二级结构间的距离越小。设两棵 Rs 树的最大节点数为 *r*，最大抽象子结构之和为 *m*，则需要遍历 *m*+*n* 个节点，故算法 3.2 的算法时间复杂度为 O(*r*+*m*)。空间复杂度为 O(*r*+*m*)。

3.3 基于 Rsd-bp 距离的多个 RNA 二级结构距离计算

3.3.1 归一化思想

归一化思想是一种无量纲处理手段，可以将计算结果的绝对值变成某种相对值关系。从而达到简化计算，缩小量值的效果。简单的说归一化思想就是把数据映射到 0~1 范围内，将有量纲表达式变成无量纲表达式成为标量，使不同单位或量级的指标能够进行比较和加权。归一化思想的实质就是一种线性变换，在这种线性变换

下，不会改变原始数据的数值排序，使衡量标准不同的数据可以表示在同一尺度范围内表示。

计算如公式(3-4)所示。

$$xn = \frac{x - \min}{\max - \min} \tag{3-4}$$

式中 x——数据集合内数据；

xn——数据归一化后的值，$0 \le xn \le 1$；

min——数据集合内的最小值；

max——数据集合内的最大值。

3.3.2 Rsd-bp 算法设计

Rsd-bp 算法是基于形状-碱基距离计算算法，通过将 Rsd 距离与 BP 距离相结合的方式，通过计算 RNA 二级结构集合中两两结构间形状距离和碱基对距离的平均值，体现 RNA 二级结构在集合空间内的特征。

设有一个大小为 n 的 RNA 二级结构样本集合 Structures，Structures={S_1,S_2..S_n}，S_i 为 RNA 二级结构，i 为结构编号。算法流程图如图 3-4 所示。

首先，对数据进行预处理，读取 Structures 中每个 S_i 的 CT 文件，本文选用的 5 列 CT 文件，保留其第 1 列和第 4 列即碱基列，去除其他多余列，使数据一行表示一个碱基和 0 或一对配对碱基对。并按升序对碱基对排序，去除无配对碱基行和重复碱基对行，进而去除无关信息，减少输入数据。

然后，根据 BP 算法和 Rsd 算法分别计算 RNA 二级结构集合中两两二级结构间的距离。由于集合中结构距离具有对称性，即结构 S_i 到结构 S_j 的距离与结构 S_j 到结构 S_i 的距离相等，所以只计算 S_i 到 S_j 的距离，$i<j$。将这些距离按照顺序依次存储在线性数组中，得到长度为 $n^2/2-n$ 序排列的距离数组。

接着，对 BP、Rsd 距离数组中的各个值进行归一化，将距离值映射在[0,1]之间，归一化后的距离值在空间集合内的数值排序不变。根据公式(3-5)计算 S_i 到 S_j 的 Rsd-bp 距离存储在数组中。

$$Rsd-bp(S_i,S_j) = \frac{xn(BP(S_i,S_j)) + xn(Rsd(S_i,S_j))}{2} \tag{3-5}$$

通过归一化的方式将 Rsd 距离和 BP 距离映射到同一区域中，计算二者的平均值，可以从形状和碱基对两方面综合衡量碱基对与形状对 RNA 二级结构间距离的影响。

使 Rsd 算法和 BP 算法产生互补效应，更好区别 RNA 二级结构之间的差异。

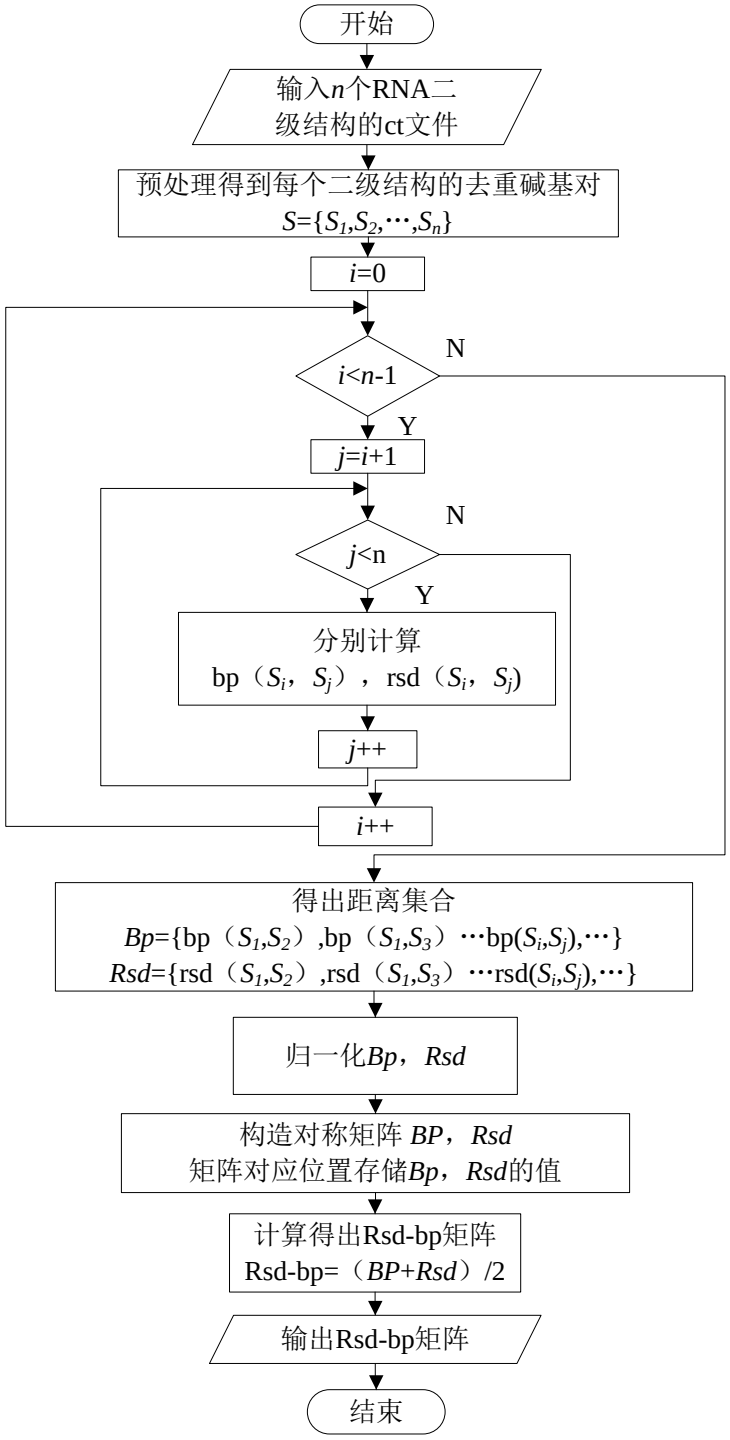


图 3-4 Rsd-bp 算法流程图

最后将 Rsd-bp 距离数组转化为对角线为零的对称评分矩阵，其大小为 $n \times n$，矩阵中的第 i 行第 j 列的元素表示第 i 个 RNA 二级结构到第 j 个二级结构的 Rsd-bp 距离。对角线上的元素表示二级结构与其自身的 Rsd-bp 距离。

3.3.3 多进程优化 Rsd-bp 算法

多个 RNA 二级结构计算的 Rsd-bp 算法属于计算密集型算法，计算时间较长。上一节的描述可知，在 Rsd-bp 算法计算过程中，需要读入 *n* 个 RNA 二级结构数据，并计算 *n* 个结构间的距离构成距离矩阵。在这一过程中 RNA 二级结构间距离的计算互相独立的，其输出结果格式相同，存储位置固定。也就是说，这一过程只有输入输出不同，运行的计算函数相同。所以为了节约计算时间，同时计算多个 RNA 二级结构间的距离，提升算法效率。

在单进程计算的计算过程中 CPU 只运行这一个任务，会造成 CPU 资源闲置，增加计算时间。随着计算机硬件水平的提升，多核 CPU 的普及使用多进程计算，并行处理任务，可以有效的提升算法效率节约计算时间。所以采用多进程优化 Rsd-bp 算法。并行执行算法中的计算函数，从而达到降低计算时间的需求。优化 Rsd-bp 算法，基本过程如下。

(1)将需要计算的 RNA 二级结构路径存于二维数组中，每一行表示需要计算的两个 RNA 二级结构。定义这个数据为 namelist[n]。

(2)构建进程工作函数，即在每一个进程上运行的计算函数，定义该函数名为 job()。

(3)根据运行环境的硬件水平创建进程池，设置进程池内的进程数 k。进程数的设定应考虑运行环境机器的 cpu 的内核数，尽量在设置进程数时低于总内核数。因为饱和的进程数会增加系统负荷，降低运算速率，影响算法的整体效率。

(4)设定进程池后，按进程数将整体运算任务平均分配到每个进程中去。使用 pool.map 函数，分配线程。Map 函数可以将输入数据按序迭代放入进程池分配给子进程。并将计算得出的值，以同样的顺序存储在固定位置上。从而保证计算值顺序的一致性。

(5)将计算结果归一化，进行变换后形成距离矩阵输出。算法描述如算法 3.3 所示。

算法 3.3: Rsd-bp 距离计算算法

输入：RNA 二级结构集合 namelist。

输出：RNA 相似度距离矩阵 M。

BEGIN

1: $S[0]\leftarrow$namelist,$S[1]\leftarrow$namelist

2: function job(*S*) #进程计算函数计算 RNA 二级结构的 BP、Rsd 距离。
3: *s1*←*S*[0],*s2*←*S*[1]
4: *bp*←*s1*∪*s2*-*s1*∩*s2*
5: *rsd*←function Rsd(*s1*,*s2*) #Rsd 距离计算函数。
6: return *bp*,*rsd*
7: function multiprocess() #进程函数。
8: create multiprocess pool←*k* #创建进程池，进程数 k。
9: put *S* in pool ,use *k* cores run function job(*S*) #将 S 放入进程函数。
10: bp,rsd←get function multiprocess() #执行多进程。
11: function xnor(*d*) #归一化函数
12: temp←[ ]
13: for i ←0 to length(*d*)
14: temp[i]←(*d*[i]-max(*d*))/(max(*d*)-min(*d*))
15: return *temp*
16: *bprsd*←(function xnor(*bp*)+function xnor(*rsd*))/2 #归一化结果
17: *M*←tansforme *bprsd* to matrix #转化为距离矩阵
18: return *M*
END

算法 3.3 使用多进程计算 RNA 二级结构的 Rsd-bp 距离，可以实现 BP 距离、Rsd 距离的并行计算降低单进程算法的时间复杂度。设输入的 RNA 二级结构集合内有 N 个样本，则在单进程计算 Rsd-bp 距离需要进行 $N^2/2-N$ 次计算，则单进程 Rsd-bp 算法的时间复杂度为 $O(N^2/2-N)$。由算法 3.3 可知，多进程 Rsd-bp 距离计算算法根据进程数并行执行函数。若设进程数为 k，则同时进行 k 次计算，则多进程 Rsd-bp 算法的时间复杂度为 $O((N^2/2-N)/k)$。多进程算法的空间复杂度与单进程空间复杂度相同，为 $O(N)$。

3.4 算法分析

综上所述，Rsd-bp 算法是一种复合算法，包含单组 RNA 二级结构碱基对与形状距离计算算法，和多个 RNA 二级结构集合内距离计算，将该算法分析划分为 3 个主要部分，bp 算法、Rsd 算法和整体算法。

(1)BP 算法，由 2.1 节可知 BP 算法的时间复杂度取决于比对的两个 RNA 二级结构碱基对的多少，所以其时间复杂度为$O(n)$，n 为碱基对对数。

(2)Rsd 算法，由 3.1、3.2 分析可知，Rsd 算法的时间复杂度为 O($n$log$n$+m+r)，n 为最大碱基对对数，r 为 Rs 树的节点数，m 为抽象结构数。

(3)对于整体算法而言，设进程数为 k，需计算$N^2/(2-N)$个二级结构之间的距离才能构成对称的 RNA 二级结构距离矩阵。故多进程 Rsd-bp 算法时间复杂度为(O(n)+O($n$log$n$+m+r))×(N^2/2-N)/k。

3.5 本章小结

本章首先对目前流行的 RNA 二级结构距离计算算法存在的缺陷进行分析，然后提出基于 RNA 二级结构形状-碱基对距离计算算法。该算法第一步是计算 RNA 二级结构的形状距离，将 RNA 二级结构抽象为带符号的有序树，通过转换、删除的树编辑操作来计算 RNA 二级结构的形状距离；第二步是根据归一化思想计算形状距离与碱基对距离的平均分；第三步是对 Rsd-bp 算法进行多进程优化，实现计算效率的提升。最后，对该算法进行算法分析。

第 4 章 基于半监督学习的 RNA 二级结构聚类算法研究

4.1 问题分析

RNA 次优二级结构集合内包含大量相似的 RNA 二级结构，在聚类之前无法判断集合内是否存在噪声点。传统的 k-medoids 算法具有较强的鲁棒性，对噪声不敏感，但该算法仍具有一些缺陷，无法满足 RNA 二级结构聚类的要求。并且传统无监督聚类算法无法充分利用计算好的 RNA 二级结构距离来提升聚类算法性能，故本文提出半监督的 k-medoids 聚类算法(SS-medoids^{+}算法)，通过从已计算的距离矩阵中提取监督信息作为先验知识，初始化中心点并引导聚类算法进行数据划分，提升聚类算法的聚类质量和效率。

4.1.1 RNA 二级结构半监督聚类存在缺陷

在 RNA 二级结构的半监督聚类算法中，由于约束集合并非根据已有先验知识获得，而是通过对 RNA 二级结构间距离计算获得数据间约束关系构成。所以在 RNA 二级结构的约束集合中可能存在一些导致聚类失败的异常约束关系，称这些约束关系为约束冲突。

约束冲突可以分为三个类型，具体定义如下。

第一类约束冲突，顺序敏感型冲突。这类约束冲突是由数据划分顺序的先后产生的。

如图 4-1(a)所示，数据 $(x_i, x_k), (x_j, x_k) \in CL$，$i < j < k$。图中用虚线连接表示数据间存在勿连约束关系，即在聚类过程中 x_i 与 x_k、x_k 与 x_j 不能划分为同一簇。假设需要将数据划分为两类 C_1 和 C_2，根据数据划分规则 x_i、x_j，分配给了 C_1 和 C_2。当分配 x_k 时发现，x_k 对于 C_1 和 C_2 中的数据存在勿连约束，故无法划分 x_k 至任何簇内，聚类失败。但若调整数据划分顺序，如图 4-1(b)所示，先划分 x_i、x_k 则不会出现冲突。同理，这类冲突也存在于必连集合内。如图 4-1(c)所示，数据 $(x_i, x_k), \left(x_j, x_k\right) \in \mathrm{ML}$，图中用实线表示数据间存在必连约束关系，即在聚类过程中 x_i 与 x_k、x_k 与 x_j 必然划分为同一簇，由于约束关系具有传递性，由公式(2-6)可以推出，x_i、x_k、x_j 为同一簇内，故按照顺序划分数据时会将x_i划分到与之较近的 C_1 簇内，则 x_k、x_j 也必须划分到$\mathrm{C_1}$中，但实际上 x_k 与和 C_2 更为接近。这时虽然不会导致聚类失败，但会增加聚

类收敛的迭代的次数，降低聚类的效率。如图 4-1(d)所示若先将 x_j、x_k 划分到 C_2 簇中，则会取得更好的聚类结果。

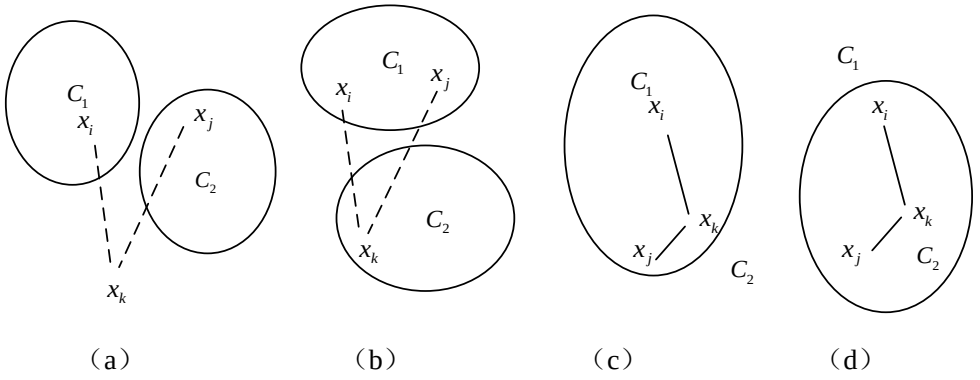


图 4-1 第一类约束冲突及调整方法

第二类约束冲突，类别敏感型冲突。这类约束冲突是由于聚类指定类别数较少而约束规则较多导致的聚类失败。

如图 4-2 所示。数据 $(x_i,x_j),(x_i,x_k),(x_j,x_k)\in CL$，故，$x_i$、$x_j$、$x_k$ 互为异簇。设划分聚类类别数为 2，此时当满足约束关系时 x_i、x_j、x_k 需要被划分到 3 个不同的簇中，与原有划分类别发生约束冲突，聚类失败。

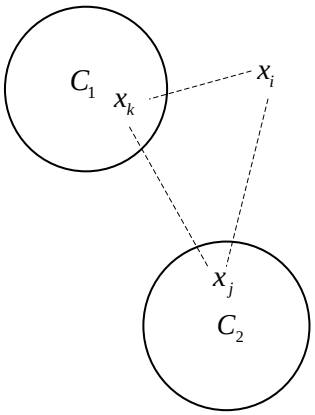


图 4-2 第二类约束冲突

第三类，矛盾型约束冲突。此类冲突是由于选定约束集合内包含错误约束关系，导致勿连集合与必连集合内的约束互为矛盾或不符合实际划分情况。

如图 4-3（a）所示数据 $(x_i,x_k),(x_j,x_k)\in ML$，$(x_i,x_j)\in CL$，此时，将 x_i、x_j、x_k 之间存在必连关系但无法划分为同一簇，数据划分过程中出现了矛盾，聚类失败。如图 4-3（b）所示数据 $(x_i,x_j),(x_j,x_h)\in ML$ 根据约束集合传递性可知 $(x_i,x_h)\in ML$ 且 $(x_i,x_k)\in CL$。但此时 x_h 到 x_i 之间的距离远远小于 x_k 到 x_i 的距离，说明在划分不合理，

存在约束冲突，聚类失败。

综上所述，第一类约束冲突，造成的聚类失败或数据划分不合理的原因是带有约束关系的数据划分先后顺序导致划分异常。说明聚类中划分带约束的数据具有顺序敏感性。因此，对于此类冲突可以通过调整数据划分的先后，降低冲突发生的可能。第二类约束冲突是由于划分类别过少，约束关系较多。需要在选择约束关系时根据需要的划分的类别数进行筛选，排除不满足类别需求的约束关系。第三类约束冲突是由于约束集合中存在错误的约束关系。需要对约束集合进行检查，排除异常约束。

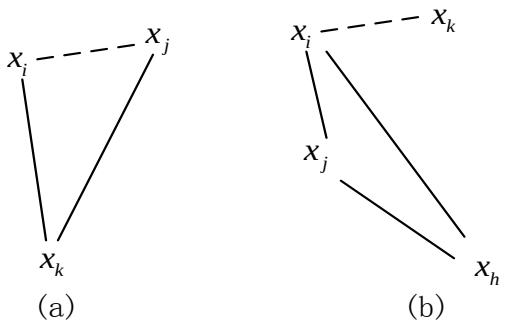


图 4-3 第三类约束冲突

4.1.2 k-medoids 算法存在缺陷

(1)中心点初始化选取不合理。传统 k-medoids 聚类算法初始化中心点由随机选择。受其随机性影响，中心初始化的选择会影响聚类结果。对于 RNA 次优二级结构集合而言，数据集内含有大量距离较近次优二级结构，若在中心初始化时，选择的中心点距离较近，容易使算法陷入局部最优解，降低算法的效率。

(2)中心点更新效率低。k-medoids 算法通过计算各个样本作为替换中心时会产生最小绝对误差，来选择新的中心点。也就是说在替换中心点时需要对未选择为中心点的所有数据进行全局查找，计算每一个数据点作为替换中心的最小绝对误差。这需要极高的开销和算法时间复杂度。

所以，为了获得更高质量的聚类结果，结合半监督学习思想，针对 k-medoids 算法的中心点初始化和更新策略进行改进。

4.2 输入数据预处理

SS-medoids^{+}算法要对 RNA 次优二级结构进行聚类分析，首先需要输入无标签的

RNA 次优二级结构集合作为数据集，以及少量包含约束关系标签的 ML，CL 集合作为强制约束，“诱导”无标签数据划分到合适的簇中。

4.2.1 数据集合预处理

数据集合即 RNA 次优二级结构集合，使用 Sfold2.0 工具中的 Srna 模块[23]折叠产生 1000 个次优 RNA 二级结构作为输入数据。因为 Srna 模块内嵌算法为玻尔兹曼抽样算法，结合了二级结构特征和热力学参数，可以统计生成一个具有代表性的样本。文献[22]证明波尔兹曼算法生成的 1000 的结构样本足以保证包括碱基对频率在内的统计重现性。此外，不管序列长度如何，预期在 1000 个结构样本中都有可以产生一个可观概率的簇。

当获得 1000 个样本集合的数据集后，使用 Rsd-bp 算法进行 RNA 二级结构距离计算。经过 Rsd-bp 算法计算后会生成一个 1000×1000 的距离矩阵，构成特征空间 $\mathbb{R}^{n\times n}$。如公式(4-1)、(4-2)所示。

$$\mathbb{R}^{n\times n}=\{\vec{s_1},\vec{s_2},...,\vec{s_n}\} \tag{4-1}$$

式中 $\vec{s_i}$ ——数据集中第 i 个二级结构的特征向量，即第 i 个结构与数据集中所有结构的距离。

$$\vec{s_i}=(s_{i1},s_{i2},...s_{ij},...,s_{in}) \tag{4-2}$$

式中 s_{ij} ——第 i 个结构与第 j 个结构的 Rsd-bp 距离。

由于 RNA 二级结构之间的距离具有对称性即 $s_{ij}=s_{ji}$、s_{ii}=0，故$\mathbb{R}^{n\times n}$为主对角线上元素均为 0 的对称矩阵。

4.2.2 约束集合预处理

本算法使用的约束集合并非实际生物实验中产生的真实数据，而是对 RNA 二级结构间距离与聚类类别数进行学习，获得的强制约束关系。要求产生的约束集合不存在导致约束冲突的异常约束。因此，首先根据 RNA 二级结构间 Rsd-bp 距离进行初步选择，然后删除导致约束冲突产生的错误约束关系，最后将约束关系转化为监督信息，监督聚类划分过程。

(1)初步选择

首先根据 Rsd-bp 距离初步选择约束集合，取$\mathbb{R}^{n\times n}$的下三角矩阵(不包含对角线上的元素)上的值构成列表 R'，如公式(4-3)所示。

$$R' = \left\{ \text{Rsd-bp}\left(s_i, s_j\right) \right\}_{N'} \tag{4-3}$$

式中 R'——数据中所有 RNA 二级结构间的单向 Rsd-bp 距离；

N'——数据集合内单向关系数量和，$N' = \frac{(n^2 - n)}{2}$，n 为数据集合大小。

然后对R′进行升序排序，定义ε为选取约束集合的比例系数，$\varepsilon<1$。$\varepsilon \times N'$等于选取约束集合的初始个数。如公式(4-4)、(4-5)所示。

$$\begin{cases} \max\_ml = R'_{\varepsilon} \\ \min\_cl = R'_{N'-\varepsilon} \end{cases} \tag{4-4}$$

式中 $\max\_ml$——ML 约束集合中约束数据对之间的最大距离；

$\min\_cl$——表示 CL 约束集合中最小距离。

$$\begin{cases} ML = (s_i, s_j) \mid Rsd - bp\left(s_i, s_j\right) < \max\_ml \} \\ CL = \{(s_i, s_j) \mid Rsd - bp\left(s_i, s_j\right) > \min\_cl \} \end{cases} \tag{4-5}$$

即强制约束 RNA 二级结构间距离大于一定阈值范围内的数据对为勿连约束，小于一定阈值范围内数据对为必连约束。

(2)ML，CL 预处理

首先对数据间约束关系有如下定义。

定义 4-1：连通量 设 x_i、x_j、x_k 为数据集中的样本数据，若(x_i,x_j)属于某类约束集合，则说明 x_i 到 x_j 是连通的。称 x_j 为 x_i 的连通量。若(x_i,x_j)、(x_j,x_k)属于同一约束集合，存在同种约束关系，则 x_i 到 x_k 也是是连通的，称 x_k 为 x_i 的间接连通量。

定义 4-2：连通度 若数据 x_i 到 x_j 是连通的，则称 x_i 到 x_j 之间经过连通量个数的总和为 x_i 到 x_j 的连通度。

定义 4-3：同类闭包 若在数据集合中，存在若干数据，数据间包含含必连约束关系且互相连通，则称这些数据构成的集合为同类闭包。

如图 4-4 所示，$(x_h,x_j),(x_j,x_i),(x_i,x_k)\in ML$，$x_h$、$x_j$、$x_i$、$x_k$ 构成同类闭包。

定义 4-4：闭包中心，若存在同类闭包 M，$M = \{x_h, x_j, x_i, x_{k,\cdots}\}_p$ 则称 M 内数据的均值中心为闭包中心 M'，如公式(4-6)所示。

$$M' = \frac{1}{p}\sum_{i=1}^{p} x_i \tag{4-6}$$

定义 4-5：异类闭包 若存在两个同类闭包 M_1、M_2，$M_1 = \{x_i, x_j, x_h, x_i\}$、$M_2 = \{x_a, x_b, x_c\}$。若 $(x_h, x_a) \in CL$，则称 M_1、M_2 为异类闭包。

如图 4-4 所示，约束集合具有传递性，所以 M_1、M_2 内所有数据不能划分为同簇。故 M_1、M_2 的闭包中心 a，b 之间具有勿连约束关系 $(a,b)\in CL$。

定义 4-6：孤立点 设有一数据 x_i，若 x_i 仅有含有勿连约束关系，不含有必连约束关系，则称 x_i 为孤立点。

如图 4-4 中x_l上的约束关系仅有 $(x_l,x_h),(x_l,x_c)\in CL$，故 x_l 为孤立点。

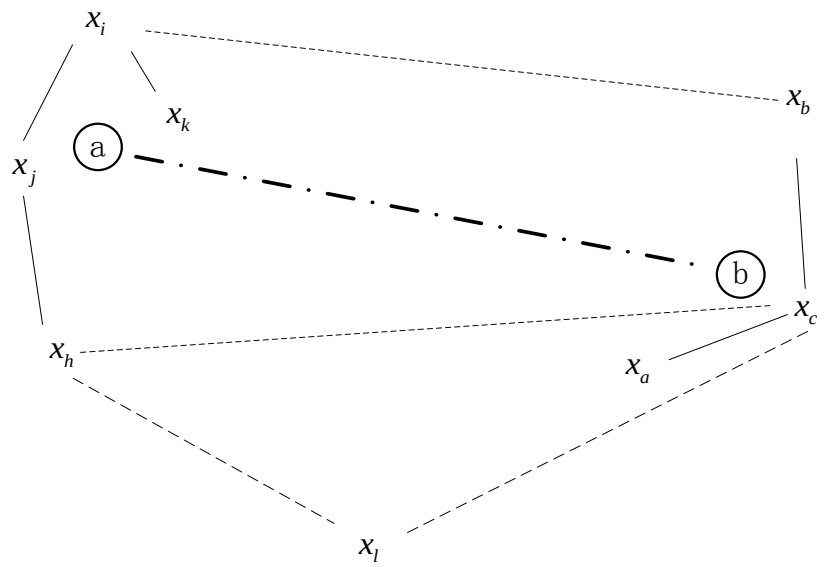


图 4-4 数据关联关系图

然后，以邻接图的形式存储数据，并对初步选择的 ML，CL 集合进行以下处理。获得同类闭包集合 neighborhoods，闭包中心列表 ce，以及闭包中心间的勿连约束关系 cl_，称这些信息为监督信息，使用监督信息代替原有的约束集合。

采用上述方式处理初步选择的 ML，CL 集合的原因有以下几点。

(1)由定义可知，孤立点在约束集合中只存在 CL 约束关系，不存在 ML 约束关系。这种点往往是距离所有数据较远的单个数据。由于 k-medoids 算法对孤立点并不敏感，使用孤立点的约束关系作为约束聚类的条件，不仅可能产生第二类约束冲突，而且对聚类划分并没有积极影响。故在处理时优先排除孤立点。

(2)使用闭包中心的方式代替部分数据，闭包中心的勿连关系替代原有数据的勿连关系。简化了约束关系，可以避免在聚类过程中因划分顺序产生的第一类约束冲突。

(3)通过调整 cl_中为环，当环上节点大于 k 时，删除该路径中点与点之间的距离最小的勿连约束，使环结构展开，可以减少形成第二类约束冲突的可能。

约束集合预处理的详细算法描述如算法 4.1 所示。

算法 4.1：约束集合预处理算法

输入：初步选择的 ML、CL，数据数目 n，聚类个数 K

输出：同类闭包中心 ce，闭包中心勿连关系 cl_，同类闭包 neighborhoods

```
BEGIN
1:  ml_graph,cl_graph←create ml_graph,cl_graph(ML,CL )
    #以邻接图的形式存储 ML，CL
2:  for i←0 to length(ml_graph)
3:      for j←0 to lengeh(ml_graph[i])
4:          if j!=i and j in cl_graph[i]
5:              return error
6:   for i←0 to length (cl_graph)
7:           if (cl_graph[i]!=null and ml_graph[i]==null)
8:               delete cl_graph[i]
9:   DFS(ml_graph)← ml_path
10: neighborhoods←ml_path,ce←[]
11: for i←0 to length(neighborhoods)
12:     ce.append(mean(neighborhoods))
13: cl_←[]
14: for i←0 to length(neighborhoods)-1
15:     for j←i+1 to length(neighborhoods)
16:         for k←0 to legth(neighborhood[i])
17:             if length(cl_graph[k] & neighborhood[j]))!=0
18:                 cl_.append(i,j)
19:                 break
20: DFS(cl_)←cl_path
21: for i←0 to length(cl_path)
22:         if (same number in cl_path)and (length(cl_path)>K)
23:             j← max(dist(j to else))
```

24: delete *cl_path*[*i*][*j*]

25: return *neighborhoods,ce,cl_*

END

算法 4.1 中将初步选择的 ML、CL 处理为监督信息，即同类闭包，闭包中心，和闭包中心间的勿连约束关系。将 ML 集合和 CL 集合以图的方式存储起来，含有约束关系的数据为图的点，约束关系为边，则有 ML 图和 CL 图，如图 4-4 所示。

算法 4.1 时间复杂度分析如下。

(1)3 到 5 行检查初步选择的 ML 集合中的约束关系是否存在第三类冲突，其时间复杂度为$O(2n)$，n 表示 ML 集合内的约束关系的个数；

(2)6 到 8 行检查勿连关系中是否存在孤立点，其时间复杂度为$O(m)$，m 表示 CL 集合中约束关系的个数；

(3)第 9 行使用深度优先算法将遍历 ML 图获得其路径集合即闭包路径，其时间复杂度为$O(|V|+|E|)$，V 表示在最坏的情况下的 ML 图中包含的节点数，E 表示边数；

(4)第 11 行到 19 行，对闭包路径进行处理获得闭包中心，时间复杂度为$O(2|V|^2)$；

(5)第 20 行使用深度优先算法将遍历 CL 图获得其路径和节点，其时间复杂度为$O(|V'|+|E'|)$，V'表示坏的情况下的 ML 图中包含的节点数，E'表示边数；

(6)21 到 25 行求取闭包中心的勿连关系，其时间复杂度为$O(|V'|)$；

综上所述，算法 4.1 时间复杂度为$O(2n+m+|V|+|E|+2|V|^2+|V'|+|E'|+|V'|)$在最坏情况下 E 等于 n 为 ML 图的最大边数，E'等于 m 为 CL 图的最大边数，故该算法的时间复杂度可化简为可化简为$O(n+m+|V|+2|V|^2+|V'|)$。

算法 4.1 空间复杂度：将 ML、CL 集合以图的形式存储其的间复杂度为$O(|V|+|E|+|V'|+|E'|)$。

4.3 SS-medoids$^+$算法设计

4.3.1 SS-medoids$^+$算法思想

SS-medoids$^+$算法聚类过程通过计算结构向量在空间中的距离远近来划分簇。本文使用欧式距离来计算数据间的空间距离，欧式距离是在欧几里得空间中数据与数据间的真实距离，即两点之间的直线距离。如公式(4-7)所示。

$$\mathrm{dist}\left(s_i, s_j\right)=\sqrt{\sum_{k=1}^{n}\left(s_{ik}-s_{jk}\right)^2} \tag{4-7}$$

式中 s_i——序号为 i 的 RNA 二级结构；

s_{ik}——第 i 个 RNA 二级结构的第 k 个特征。

SS-medoids$^+$算法的基本思想是根据已知的监督信息，按照距离计算公式，在满足约束信息的前提下，将距离较近的 RNA 二级结构划分为一簇，并使簇与簇之间的具有相对大的距离。其详细步骤如下。

SS-medoids$^+$算法首先，根据约束集合预处理得到的同类闭包集合和对应的闭包中心，生成 k 个初始化中心点。为使初始化的中心点分散化，防止初始化的中心点之间距离较近，影响聚类效果，要求选择的初始化中心点优先满足以下需求。

(1)优先选择规模最大的同类闭包中的数据;

(2)选择靠近闭包中心上的数据，即与闭包中心距离欧式距离最短的数据;

(3)若聚类数目 k 大于同类闭包集合个数 p，寻找 k-p 个不能连接到每个同类闭包集合的点作为补充初始化中心点。

然后，根据初始化中心点进行半监督聚类划分，SS-medoids$^+$在划分过程中分为两部分，一是对带有约束信息的数据进行划分，二是对无约束信息的数据进行划分。

(1)对于有约束信息的数据，以闭包中心替代数据。首先将闭包中心划分到距离其最近的初始化中心点代表的簇内。然后，根据闭包中心的勿连约束进行调整，将具有勿连约束的中心点划分到不同簇中。为了减少因划分顺序敏感导致的第一类约束冲突产生，需要保证所有数据在划分时尽可能划分到距离其最近的中心点簇内，且不产生约束冲突。所以在调整闭包中心之前，将这些闭包中心按其与对应中心点的距离的大小进行排序，保证距离中心点较小的点不做调整。最后，根据调整好的闭包中心，将同类闭包集合内的数据划分到其中心点对应的簇内。

(2)对于无约束信息的数据数据，则按照传统 k-medoids 算法划分规则，将数据划分至距离其最近的中心点表示的簇中，由此实现了数据的完整划分划分。

接着，根据聚类划分的结果更新中心点，提出新的中心点更新规则，减少更新中心点的搜索范围，缩短时间开销。

最后，重复迭代聚类划分和中心点更新，直到中心点不在变化，聚类算法完成。

SS-medoids$^+$算法的整体算法流程图如图 4-5 所示。

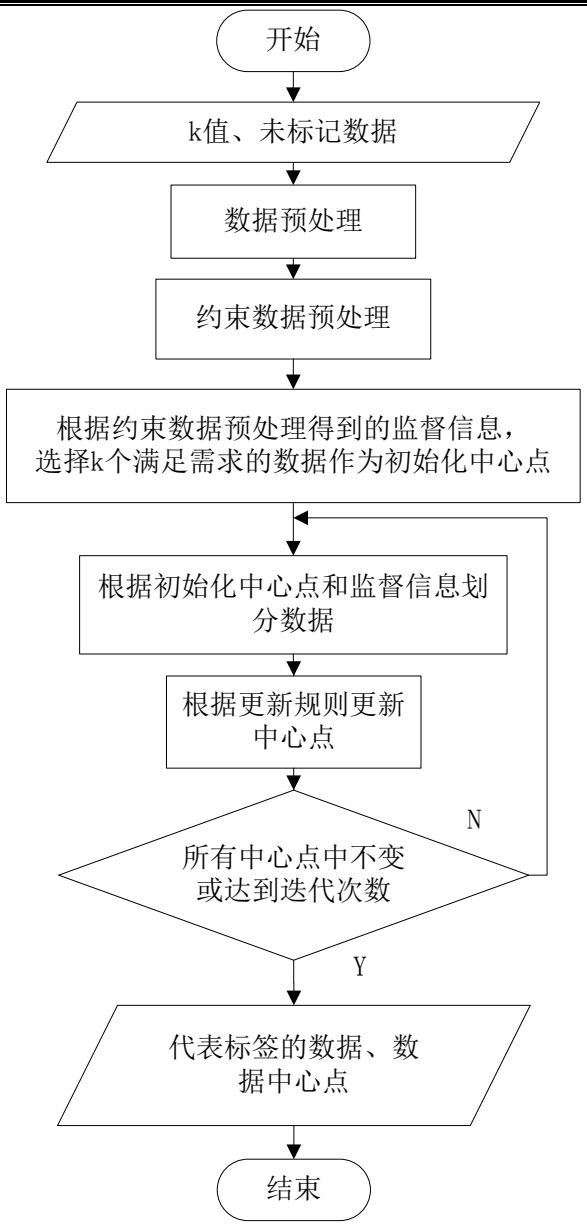


图 4-5 SS-medoids^{+}算法流程图

4.3.2 中心点更新规则

在聚类的划分过程中，需将数据划分给距离中心点最近的簇。由此可以推出，簇内最佳中心点在当前中心点附近，所以依据中心点增量方式从中心点附近的点开始查找，直到找到最优中心点为止。这种方式从一定程度上缩小了查找范围，但是在最坏的情况下其算法复杂度仍未改变仍为$O(n^2)$，所以本文提出一种可变阈值的查找方法，更新中心点，缩小了数据查询范围，提升了搜索效率。

在中心点替换研究中发现，中心点的选择依赖于中心点相对于簇内所有数据之

间的距离总和。随着中心点变化，数据间距离总和也会产生变化。在原始中心点周围存在许多次优可替换中心点，这些点对于簇内其他数据距离之和小于原中心点，但并非为全局最小。若选取其中一点，以该点为中心继续查找会查找到更优于该点的点。也就是说，在搜索中心点时可以从距离当前中心点最近距离的数据点开始查找。当查找到优于原中心点的替换中心点时，以替换中心点为中心继续查找。直到当前替换中心点阈值范围内没有优于当前替换中心点的数据时，当前中心点则为最优替换中心点。

根据这一思想定义如下。

定义 4-7：替换的代价 数据替换中心点时产生的代价称为替换代价 cost，当代价函数小于 0 时，候选中心点为优于原中心点的可替换中心点，如公式(4-8)所示。

$$\mathrm{cost}(v,e)=\sum_{i}^{n}dist(v,x_i)-\sum_{i}^{n}dist(e,x_i) \tag{4-8}$$

式中 v——原中心点；

e——候选中心点；

x_i——搜索范围内的数据；

n——簇内数据总个数。

定义 4-8：搜索阈值 以中心点为中心，搜索可替换中心点的半径称为搜索阈值。如公式(4-9)所示。

$$\begin{cases}\vartheta_0=\max(\mathrm{dist}(m0,x_i))\\ \vartheta=\vartheta_0-\mathrm{dist}(m0,v)\end{cases} \tag{4-9}$$

式中 m_0——中心点；

ϑ_0——初始搜索阈值；

V——可替换中心点；

ϑ——替换搜索阈值。

由公式(4-9)可知ϑ_0为初始中心点 m_0 据对应簇内数据的最大半径。由于最优中心点不一定只出现于本簇内，还可能出现在于与该簇较近的其它簇数据内。且原始中心点并非几何中心，所以以最大半径为搜索阈值可以有效的扩大搜索范围。而在搜索过程中可替换的中心点一定在原中心点附近，所以对其搜索半径缩小为ϑ，即当前中心点到最大阈值边界距离。

由图 4-5，表示中心点的替换过程，可以看出这种可变搜索阈值的方式可以尽可

能的减少不必要的查询，节约时间开销，提升算法效率。其算法描述如算法 4.2 所示。

算法 4.2：中心点更新算法

输入：初始中心点 *m0*，数据标签 label #*m0*=[m_1,m_2,…,m_k]

输出：更新中心点 *m*

```
BEGIN
1:  function get_m(m0)
2:      m←[]
3:      for i←0 to length(m0)
4:          clusterspace← re_index(label,i)   #获得当前簇内标签
5:          D←sum(dist(clusterspace[j],m0[i]))   #计算中心点与簇内数据总和
7:          maxd←max(dist[clusterspace,m0[i] )
            #计算簇内数据与中心点的最大半径
8:          space←seg_maxd(maxd,dist[:,m0[i]])
            #根据最大半径获取搜索空间内数据
9:          newm,D=find_m(m0[i],maxd,cluster,space,D,m0[i])
           #调用函数 find_m 查找中心点附近是否有可替换中心点
10:         m.append(newm)
11:     return m
12: function find_m(m0[i],maxd,cluster,space,D,m)
13:     if space==null
14:         return m,D
15:     else
16:         D1←sum(dist(clusterspace[j],m0[i]))
17:         m1←m,al=[]     #记录数据是否访问过
18:         while(D-D1<0 or space==null)
19:             m←space[0],al.append(m)
20:             space.pop(0)
21:             D1←sum(dist(clusterspace[j],m0[i]))
22:             if space==null #搜索半径内无可替换中心
```

23: *m*←*m1*

24: return *m*,*D*

25: else:

26: *D*←*D1*

27: *maxd1*← [*maxd* - dist(*m0*,*m*)]

28: *space*←seg_maxd(*newr*,dist(:,*m*))

29: *sapce*←*space-al*

30: *M*,*d*← f_m(*m0*[*i*],*maxd*,*cluster*,*space*,*D*,*m*)

31: return *M*,*d*

END

算法 4.2 通过可变搜索邻域的方式进行中心点替换减少了数据比对的次数提高了算法效率。

算法 4.2 时间复杂度分析主要包括两部分：(1)12 行到 31 行通过递归的方式查找某一中心点可替换中心点，其时间复杂度为$O(\log N)$，N 在最坏的情况下为该簇搜索最大半径内的数据个数；(2)1 到 11 行为主函数，遍历搜索该簇内的新中心点。综上所述，算法 4.2 的总时间复杂度为$\mathrm{O}(N\log N)$，空间复杂度为$\mathrm{O}(N)$。

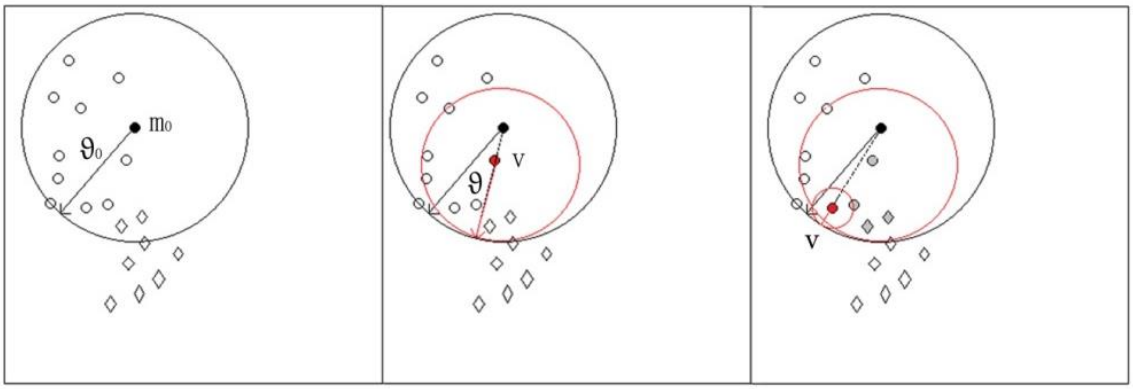


图 4-5 中心点替换过程

4.4 算法分析

在中心点初始化上，原始算法采用随机选取的方式从数据中随机选取k个数据，作为初始中心点。由于其每次初始化中心点位置不一定相同，每次聚类产生的结果也可能不同。在其最坏的情况下，若初始化中心点过于集中，则会使聚类划分陷入局部最优，降低聚类性能。而SS-medoids^{+}算法根据半监督信息初始化中心点，使初

始化的中心点之间具有一定距离，且固定不变。增强了聚类的稳定性，提高了聚类性能。

在聚类的划分上，原始算法将所有数据一一划分至与其距离最小的中心点所在簇内。SS-medoids^{+}算法加入了优化的监督信息，使用闭包中心替代部分数据进行簇的划分，通过闭包中心的勿连约束限至聚类划分，可以加快聚类收敛到最终结果，提升算法效率。

在中心点更新规则算法中，假设数据集规模为n，需要将数据划分为k个簇。原始算法，需要替换k次中心点，在每次中心点替换时需要查询x-k个非中心点是否为可替换中心点，其时间复杂度为$O(k(x-k)^2)$，x为总数据个数。由算法4.2分析可知，SS-medoids^{+}算法整体时间复杂度为$O(k\times N\log N)$，N表示簇内的最大查找数据，N<<x。所以，SS-medoids^{+}算法在中心点更新上效率有所提高。

综上，SS-medoids^{+}算法在聚类的稳定性与效率上应高于原始算法。

4.5 本章小结

本章针对传统算法采用随机选择的方式初始化中心点，易陷入局部最优解的问题，提出基于半监督思想的k-medoids聚类算法SS-medoids^{+}算法。首先分析半监督聚类应用于RNA二级结构聚类时存在的缺陷以及传统聚类算法存在的缺陷。然后根据RNA二级结构数据集合的特点提出SS-medoids^{+}算法。该算法对数据集合和约束集合进行预处理，将约束集合转化为可用的监督信息。根据监督信息进行中心点初始化和数据划分。改进原始算法的中心点更新规则，实现了RNA二级结构聚类。最后对该算法与原始算法进行对比分析。

第 5 章 实验验证

5.1 实验环境及数据集来源

实验使用 python 语言进行程序设计。实验环境为 Windows7-64 位系统，CPU 以及 RAM 内存分别为 2.5GHz 以及 8GB。

实验所用 RNA 序列源于 RNA STRAND 数据库，下载地址 http://www.rnasoft.ca/。使用 Sfold2.0 软件包[23]对序列进行折叠获得 RNA 二级结构。

RNA 二级结构和统计分析数据库(RNA STRAND)[56]提供大量不同生物和不同类型的已知 RNA 序列与其真实 RNA 二级结构。目前该数据库提供 4666 个高度准确的真实 RNA 二级结构，这些数据通过 NMR 或 X 射线晶体学实验方法确定。RNA STRAND 支持通用 RNA 序列格式搜索和下载。

Sfold 软件的 Srna 模块可根据 RNA 序列提供 RNA 二级结构采样统计数据。该模块内嵌玻尔兹曼采样算法，根据热力学参数，可以从统计学角度生成一个有代表性的 RNA 二级结构集合。

5.2 多进程优化 Rsd-bp 算法

本节分为两个部分，第一部分检验 Rsd-bp 算法对 RNA 二级结构距离计算的可行性。第二部分检验多进程优化对 Rsd-bp 算法计算时间的影响。

5.2.1 Rsd-bp 算法计算能力验证实验

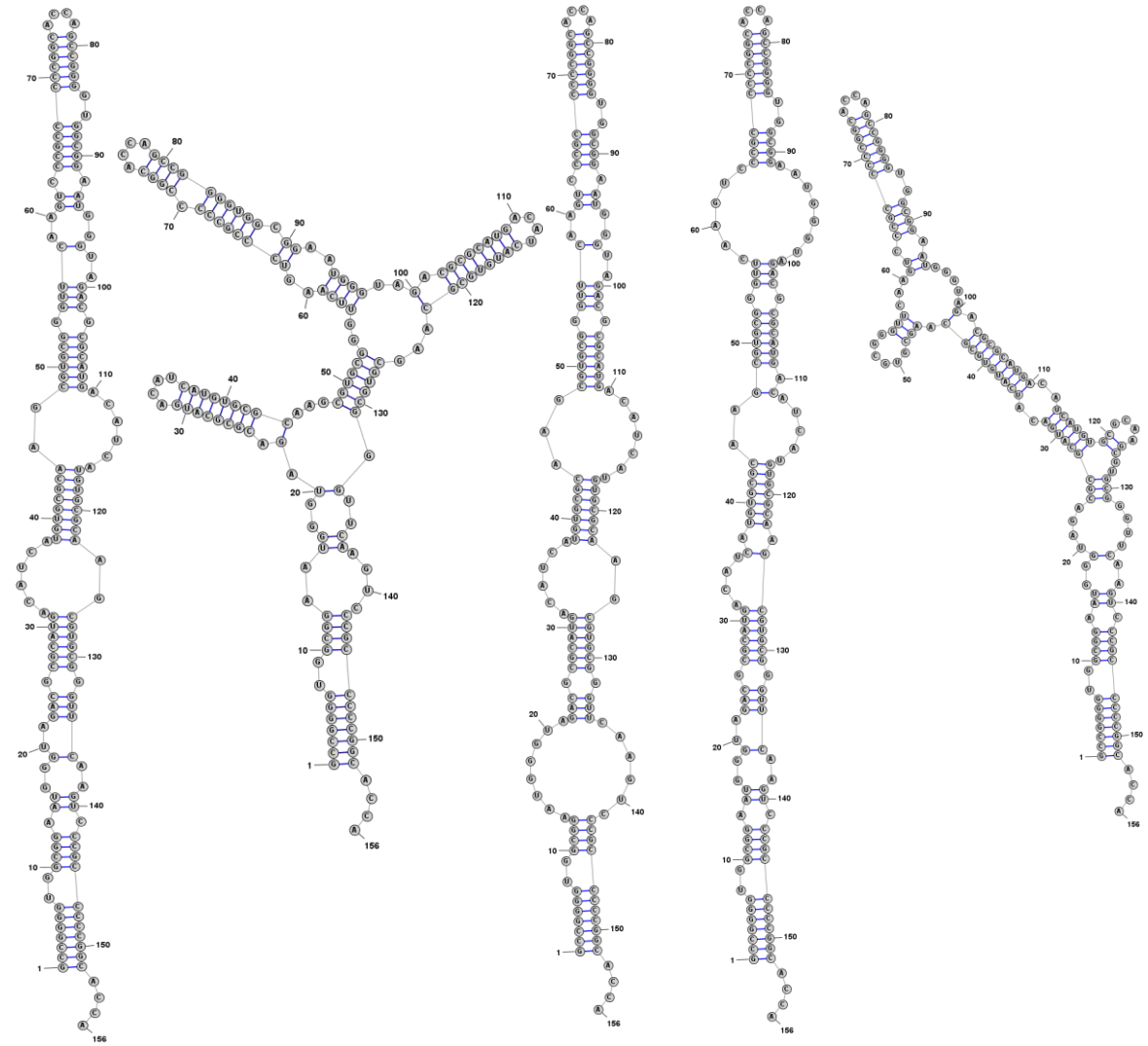
该部分实验将实验数据分别运行在 Rsd-bp 算法以文献[25,26]中使用的 BP 算法、RBP 算法。根据实验结果，验证 Rsd-bp 实验对于 RNA 二级结构距离计算的可行性和计算效率。

在数据选取方面，选取序列信息如表 5-1 所示。使用 Sfold 工具包，生成该序列的 RNA 二级结构集合。从集合中筛选 5 个具有一定特点的 RNA 二级结构作为实验样本。绘制这 5 个 RNA 二级结构的多边形图，如图 5-1 所示。通过 RNA 二级结构二维平面结构图可以直观的观察 RNA 二级结构间的形状和碱基对差异，更好的对实验结果进行分析。

表 5-1 序列信息

Database ID	Organism	type	length
PDB_00967	THERMUS THERMOPHILUS	Other RNA	156

GCCGGGGUGGCGGAAUGGGUAGACGCGCAUGACAUCAUGUGCGCAAGCGUGCGGGUUC
AAGUCCCGCCCCCGGCACCAGCCGGGGUGGCGGAAUGGGUAGACGCGCAUGACAUCAU
GUGCGCAAGCGUGCGGGUUCAAGUCCCGCCCCCGGCACCA



ΔG_{37° = -81.60　ΔG_{37° = -80.00　ΔG_{37° = -78.40　ΔG_{37° = -76.90　ΔG_{37° = -75.30

(a)S1　(b)S2　(c)S3　(d)S4　(e)S5

图 5-1 RNA 二级结构多边形图

实验步骤如下。

(1)输入数据集合对数据进行预处理和批量计算。

首先将 Sfold 生成的 ct 文件处理成碱基对格式。Sdold 生成的 ct 文件为 6 列 ct

文件。其中，提取第 1 列和第 5 列进行处理。第 1 列和第 5 列表示表碱基是否配，及配对信息。需要将其处理为无重复配对的单纯碱基对格式。数据预处理过程描述如算法 5.1 所示。

算法 5.1：数据预处理

输入：sfold 输出的 ct 文件。

输出：无重复碱基对的碱基对数组。

```
BEGIN
1:  function pretreatment()
2:      v=0                                  #记录循环次数
3:      for line to file_end                 #按列读取文档
4:       line←re.split(“s+”,line)
5:       temp←line
6:      #使用正则表达式匹配文件中的空格，按空格将数据分割为 6 列数组
7:      if temp[0]==v                #跳过文件头部无关信息
8:          a←temp[0],b←temp[4], v←v+1
9:      if ( a!=0 and b!=0)and (a<b)        #添加不重复碱基对到数组
10:         t←[a,b]
11:         add t in structure array
12:     return structure
END
```

(2)将处理好的数据输入 Rsd-bp 算法、BP 算法、RBP 算法中。对于 RBP 算法暂不考虑其最佳松弛系数，输入其默认松弛系数 1。

(3)可视化结果与分析。

将结果以图表形式表达，并依据结果分析证明 Rsd-bp 算法的可行性和优势。为了直观的对比不同算法的计算结果，对结果进行归一化处理，计算公式如公式(3-2)所示。计算结果如表 5-2、5-3、5-4 所示。

实验结果分析如下。

(1)由表 5-2、5-3、5-4 数据可以看出，Rsd-bp 算法、BP 算法、RBP 算法都对于S_1、S_3、S_4之间的距离都给出了一个较小的距离，S1、S3、S4 与 S2 之间的距离都

给出了较大的距离。从图 5-1 中也可看出 S1、S3、S4 较为相似，S2 与 S1、S3、S4 十分相异。Rsd-bp 距离与其它两种算法的结果相近，且实际情况相同，说明计算值有效。

表 5-2 BP 算法运行结果

BP 算法	S1	S2	S3	S4	S5
S1	0	1	0.076923	0.102564	0.679487
S2		0	0.948718	0.974359	0.961538
S3			0	0.102564	0.679487
S4				0	0.679487
S5					0

表 5-3 RBP 算法实验运行结果

RBP 算法	S1	S2	S3	S4	S5
S1	0	1	0.103448	0.103448	0.344828
S2		0	1	1	0.62069
S3			0	0.103448	0.344828
S4				0	0.344828
S5					0

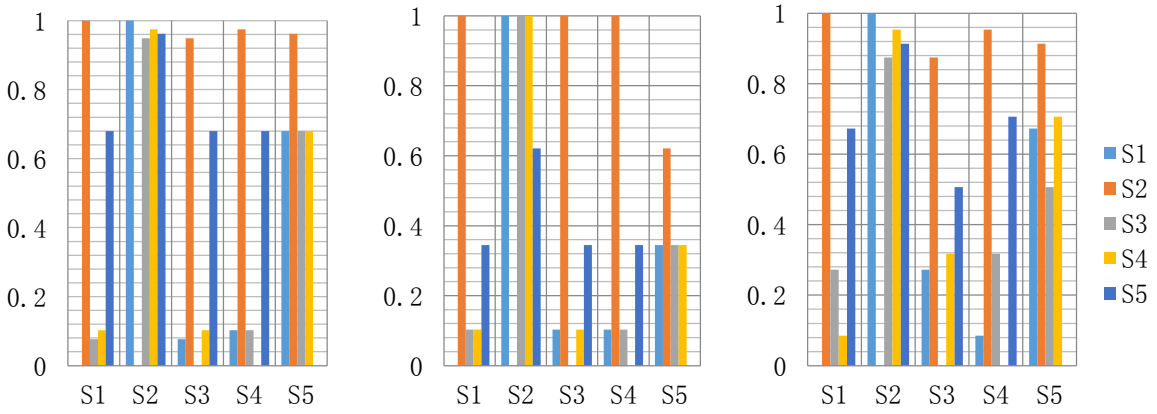
表 5-4 Rsd-bp 算法实验运行结果

Rsd-bp 算法	S1	S2	S3	S4	S5
S1	0	1	0.271795	0.084615	0.673077
S2		0	0.874359	0.953846	0.914103
S3			0	0.317949	0.50641
S4				0	0.70641
S5					0

(2)基于细节观察表格，BP(S1,S4)=BP(S3,S4)，RBP(S1,S4)=RBP(S3,S4)。两个算法对于 S1、S4 和 S3、S4 之间的距离计算相同，但 Rsd-bp 算法计算的距离相差较大，Rsd-bp(S1,S4)=0.084615>Rsd-bp(S3,S4)=0.317949。根据图 5-1 验证，可以看出 S1、S3、S4 之间确实存在很高的相似度，但对于结构 S4 而言，结构 S1 在结构 S4 碱基序号为 100 的位置多出一个内环，其他结构位置顺序基本相同，尤其在结构

S4 碱基号为 125 到 150 之间，结构 S4 与结构 S1 对应位置完全重合。而结构 S3 对于结构 S4，整体结构位置逆转，结构顺序完全不同。故 S1 与 S4 更为相似，S3、S4 之间的距离应大于 S1、S4 之间的距离。说明 Rsd-bp 算法的区分能力优于 BP 算法和 RBP 算法，可以区分 RNA 二级结构间的细小差异。

(3)将表格中的数据整理绘制成柱状图如图 5-2 所示。从整体来看，Rsd-bp 距离数值分布更为广泛，结构间的差异呈过度趋势。而 BP 距离和 RBP 距离评分值跨度较大，说明 Rsd-bp 算法可以区分结构间的细小差异，而 BP 只是粗略的计算 RNA 二级结构间的碱基距离，RBP 算法虽然弥补了 BP 距离的缺陷，但弱化了环结构与茎区的位置关系差异。如图 5-2 中结构 S5 对于各个结构的距离相比 BP 距离都有所降低，这是由于 RBP 算法的松弛系数决定了其计算结果趋于两极化。不排除本次选择松弛系数的不合理可能，但在实际研究中很难简单确定 RBP 算法的松弛系数。因此该算法十分灵活，但依赖于主观经验。较 Rsd-bp 算法而言，Rsd-bp 距离呈锯齿状分布说明在计算 RNA 二级结构距离上具有较高的区分能力，且该算法不存参数选择，客观性强，计算结果更为严谨。



(a) BP 距离柱状图　　(b) RBP 距离柱状图　　(c)Rsd-bp 距离柱状图

图 5-2 实验结果柱状图

5.2.2 计算时间验证实验

本节实验主要测试多进程优化对于 Rsd-bp 算法的改进效果，对多进程 Rsd-bp 算法、单进程 Rsd-bp 算法、BP 算法、RBP 算法进行时间测试。实验序列采用表 5-1 中序列，使用 Sfold 生成 200 个 RNA 二级结构，根据样本容量不同，对这些样本进行分组测试。分别记录单进程 Rsd-bp 算法、多进程 Rsd-bp 算法、BP 算法、RBP 算

法运行时间如表 5-5 所示。由于本实验硬件环境限制，多进程 Rsd-bp 算法在测试时使用的进程数为 4。

表 5-5 单进程 Rsd-bp 与多进程 Rsd-bp 算法执行时间

样本容量	BP	单进程 Rsd-bp	多进程 Rsd-bp	RBP
20	0.328278194s	0.248456754s	2.724564716s	120.4398361s
40	0.616470787s	1.085086355s	3.018546046s	473.9300844s
60	1.247749447s	2.423826447s	4.257824338s	1045.406716s
80	2.214789916s	4.302263135s	5.522502655s	1877.810757s
100	3.363923501s	6.636918468s	7.159500763s	2949.709778s
120	5.226140691s	9.621371395s	9.227091447s	3239.13954s
140	8.173781803s	14.59179944s	11.37150779s	3886.123485s
160	8.84664015s	17.36216198s	15.069188s	4533.107431s
180	10.89827817s	22.49073978s	17.14743241s	5180.091376s
200	16.52766818s	27.75576088s	21.5747778s	5827.075322s

据表格 5-5 分析，RBP 算法时间开销最大，是其他算法时间开销的几十倍。BP 算法由于其计算简单，计算时间最小。单进程 Rsd-bp 算法和多进程 Rsd-bp 算法在计算时间的增长趋势随着样本容量的增加相交于一点后差距逐渐拉大，二者的计算时间的趋势图如图 5-3 所示。

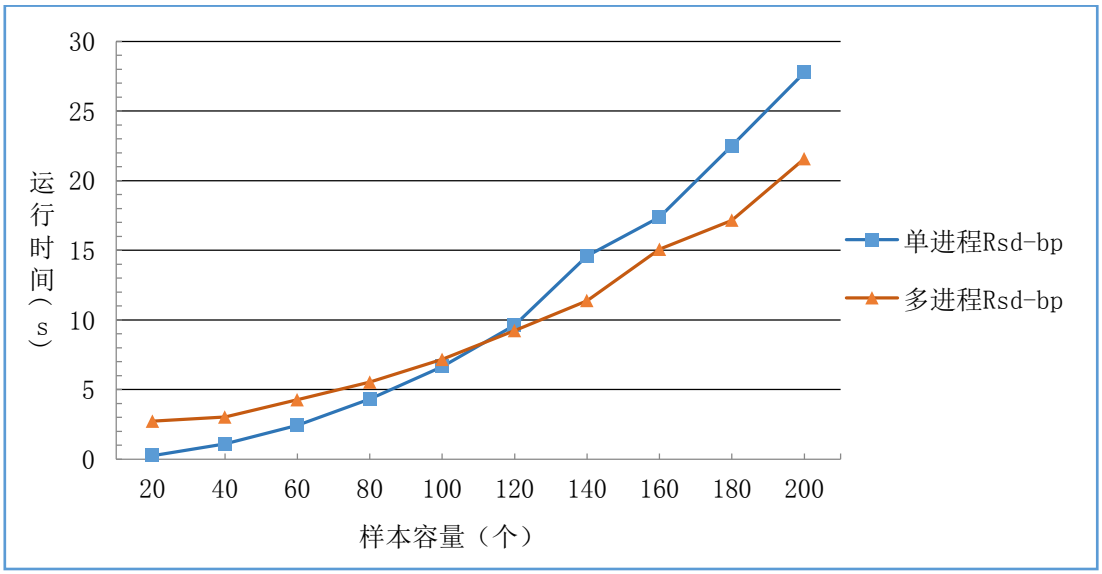


图 5-3 单进程 Rsd-bp 算法与多进程 Rsd-bp 算法运行时间对比

由图 5-3 可知，当样本数量小于 100 个时单进程 Rsd-bp 算法计算时间要低于多进程 Rsd-bp 算法计算时间。当样本容量大于 100 时多进程 Rsd-bp 算法的计算时间低于单进程 Rsd-bp 算法。从二者的趋势来看，随着样本容量的增大，单线程 Rsd-bp 算法的运行时间上升速率要高于多进程 Rsd-bp 算法。这说明随着样本容量的不断扩大，多进程 Rsd-bp 算法的计算时间会远远小于单线程 Rsd-bp 算法。这是由于，进程在创建和切换时会占用一部分时间，在小容量样本中，进程创建时间大于计算时间，此时多进程计算的运行时间大于单进程计算。随着样本容量增加，计算次数增加，进程创建时间占整体时间的比例缩小，此时多进程计算的运行时间小于单进程计算。因此，多进程计算更适用于大容量样本之间 Rsd-bp 距离计算。

5.2.3 整体分析

从算法区分能力上看，Rsd-bp 算法有能力区别 RNA 二级结构的细微差别，可以从碱基对和形状两方面计算 RNA 二级结构间的距离，优于 BP 算法、RBP 算法的计算结果。

从时间效率上看，Rsd-bp 算法的运行时间上高于 BP 算法，但相比于 RBP 算法的高计算时间开销，具有很大的优势。而多进程 Rsd-bp 算法比单进程 Rsd-bp 算法在大容量 RNA 二级结构计算中更有优势。

5.3 基于 SS-medoids^{+}算法的 RNA 二级结构聚类实验

5.3.1 实验数据

实验选取 4 条长度不等的 RNA 二级结构序列，序列的详细信息如表 5-6 所示。使用 Sold 软件将序列折叠分别生成 1000 个 RNA 二级结构进行测试。

表 5-6 实验使用 RNA 序列信息

ID	Database ID	Organism	Type	length	$\overline{M_{bp}}$
1	PDB_00052	Homo sapiens	Transfer RNA	76	32
2	PDB_00967	THERMUS THERMOPHILUS	Other RNA	156	55
3	CRW_00438	Homo sapiens	16S rRNA	954	451
4	PDB_01273	THERMUS THERMOPHILUS	16S Ribosomal RNA	1496	507

5.3.2 评分标准

本实验采用常用的聚类评价指标 CH 系数，作为衡量聚类结果好坏的算法评分标准。CH 系数(Calinski-Harabaz Index)是一种聚类质量的评分方式。表示聚类的分离度和紧密度之比，如公式(5-1)所示。CH 值越大表示簇内数据越紧密，簇间距离越分散，聚类的效果越好。

$$CH\left(K\right)=\frac{\mathrm{B}\left(K\right)}{K\text{-}1}\times\frac{\left(N\text{-}K\right)}{\mathrm{W}\left(K\right)} \tag{5-1}$$

式中 K——类别数；

B(K)——簇间距离平方和；

W(K)——簇内距离平方和。

5.3.3 实验步骤与结果分析

(1)距离矩阵计算。

首先将序列输入到 Sfold 中 Srna 模块进行序列折叠，折叠个数设定为 1000。然后将折叠好的结构打包输入到 Rsd-bp 算法中，计算 Rsd-bp 距离形成 1000×1000 的距离矩阵。

(2)数据输入和参数选取。

将(1)中得出数据输入到 SS-medoids^{+}算法中对算法进行参数测试。

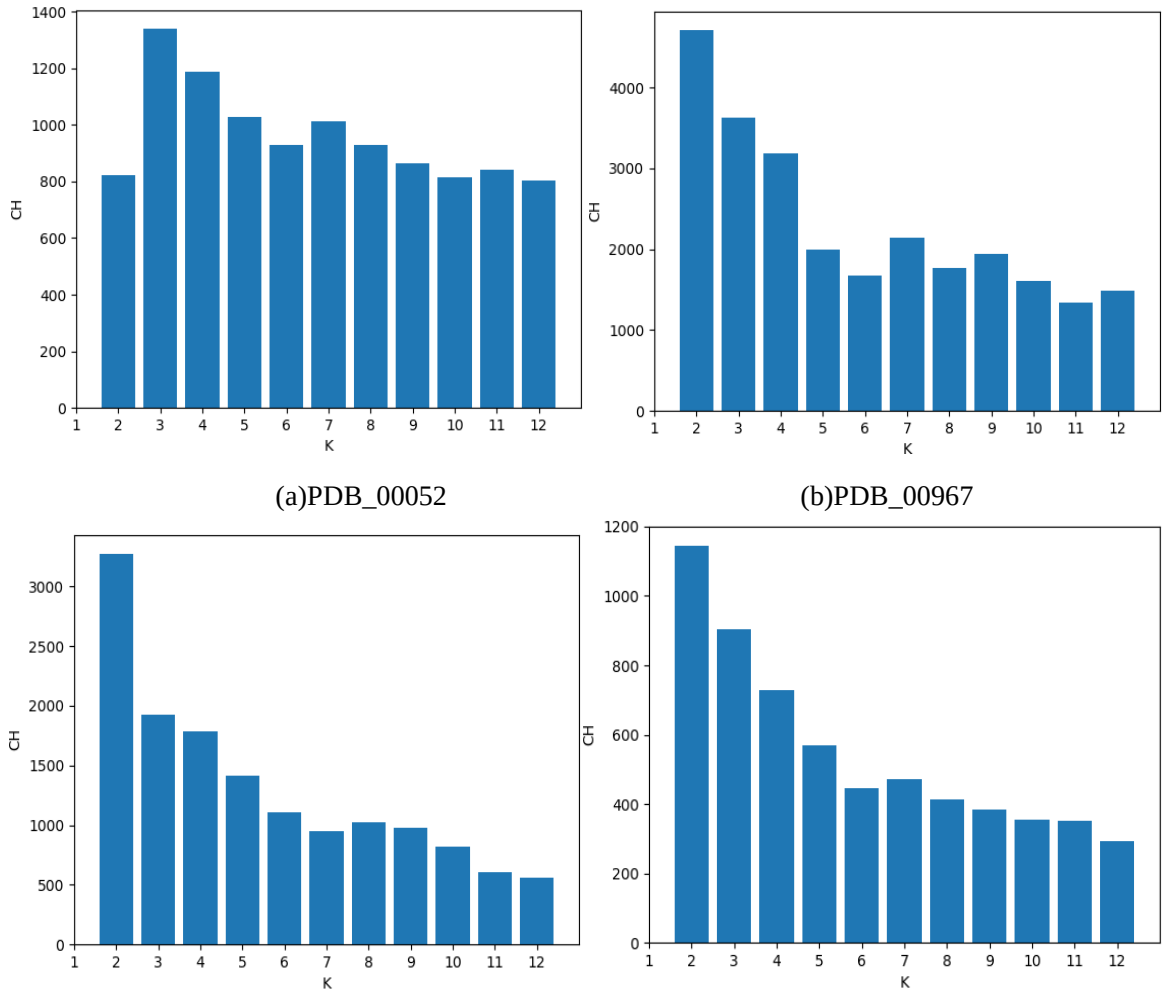
首先确定 k 值。使用 CH 系数对聚类效果进行评估，CH 系数越大表示聚类的效果越好，k 值的选择越合理。所以，选取在闭合区间[2,14]内选取 k 值进行重复测试。测试结果如图 5-4 所示。

由图 5-4 可知，当 CH 值最大时，数据集聚类对应的 k 值为最优值。故序列 PDB_00052，PDB_00967，CRW_00438，PDB_01273 的数据集聚类所选择的最优 k 值依次为 3、2、2、2。

然后对比例系数ε进行测试。比例系数ε是约束关系占整体数据总关系的百分比，其变化与数据容量的大小有关。由于本文选取数据集均为 1000 个 RNA 结构，所以在不同数据集间的比例系数是一致的。

首先选取序列 CRW_00438 作为测试数据，其余序列数据为验证数据。设定 k

值为该序列最优 k 值 2，选取$\varepsilon = [0.0001,0.5]$之间按 0.02 的间隔进行选取数据进行测试，测试发现当ε>0.05时产生第三类约束冲突，需要程序修改ε值的范围。故将测试的阈值范围调整至[0.0001,0.05]之间，重新测试。分析ε值对聚类 CH 值和聚类时间 t 的影响。

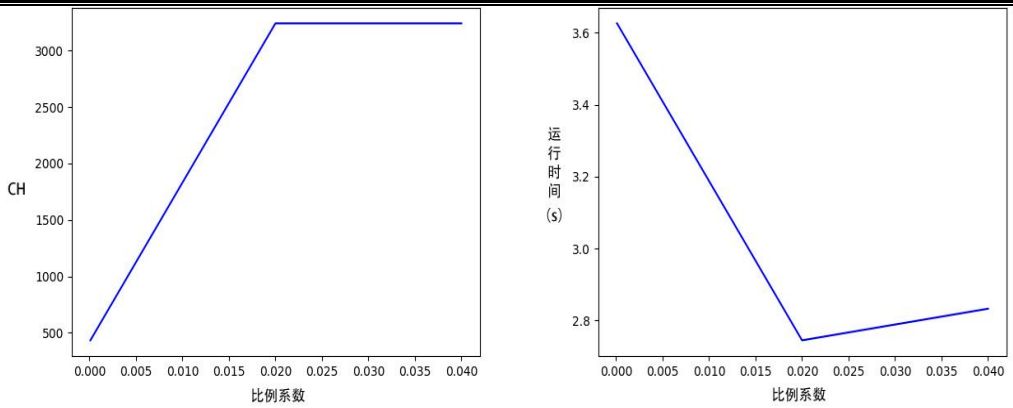


(a)PDB_00052　　(b)PDB_00967

(c)CRW_00438　　(d)PDB_01273

图 5-4 测试数据 CH 索引图

如图 5-5 所示，随着ε的增大 CH 值逐步稳定，说明算法可以通过对约束集合的预处理得到合适的监督信息，进行算法的半监督学习。由图 5-5(a)可以看出，运行时间随ε值的增大而降低直到ε=0.02 时运行时间开始增加，但增加的幅度很小。由于 ε 为分数，取值范围较大无法确定精确的阈值点，故设定ε的取值范围在 0.020 附近，可得到较好的聚类效果。使用ε=0.020 对其余序列数据进行验证，验证结果如表 5-7 所示。



(a)比例系数对 CH 值的影响趋势图　　(b)比例系数对运行时间的影响趋势图

图 5-5 比例系数取值对于算法的影响

(3)可视化聚类效果。

由于 RNA 二级结构集合为 1000×1000 的高维度非线性空间，无法直接在二维或三维空间中表示。所以需要将 RNA 二级结构集合投影到低维度空间，实现聚类效果可视化表示。

本文采用的降维算法是主成分分析法(PCA)，PCA 算法是目前应用最为广泛的一种数据降维算法。PCA 算法通过对数据进行变换获得新的坐标系。首先使数据投影的最大方差位于第一坐标轴，即第一主成分。第二大方差位于第二坐标轴，依次类推，获得 n 个坐标轴即 n 个成分。数据对于这些坐标的轴投影方向是互相正交的，这些方差值越大说明方差矩阵的特征值越大。依次，取前 k 个包含大部分方差坐标轴作为新的数据维度，实现对数据特征的降维处理。这种处理方式可以尽可能的保留数据的原始特征，降低损失率。

这里使用 PCA 算法将数据将至 2 维，绘制聚类效果图如 5-7 所示。

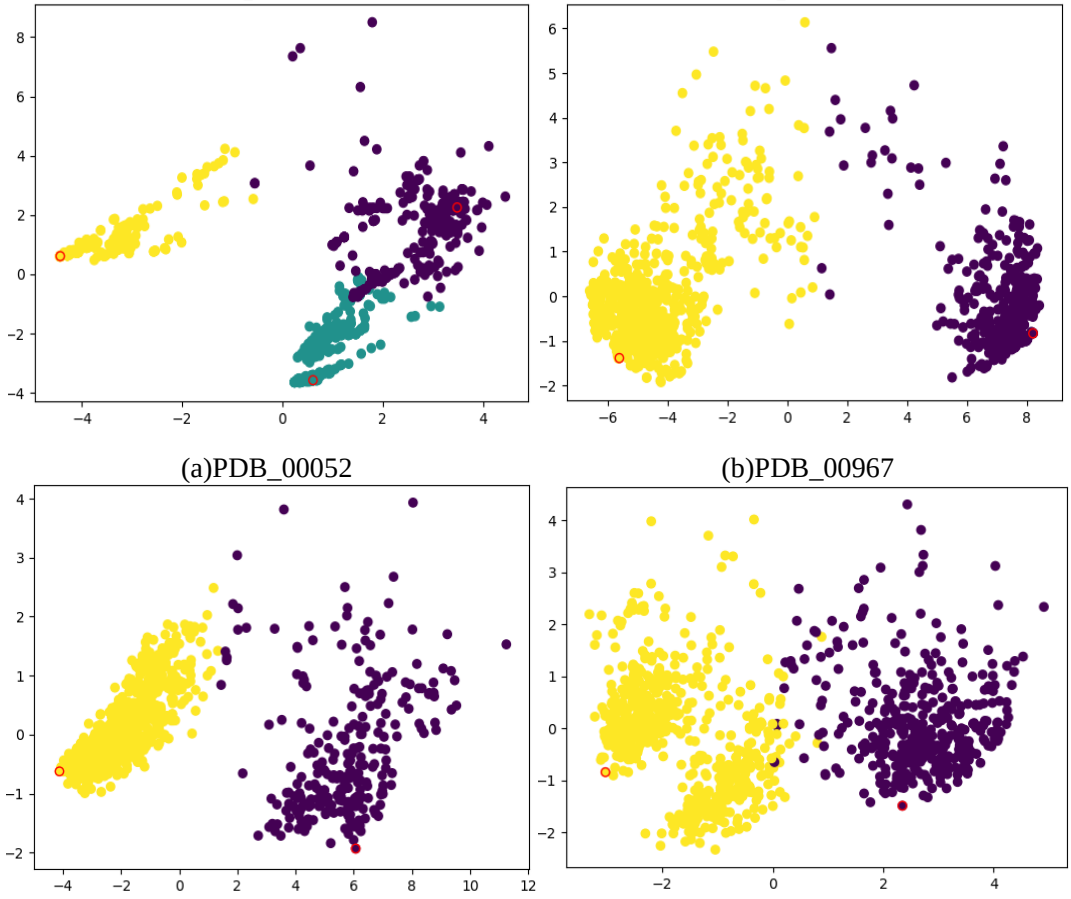
(4)性能比对分析。

计算 SS-medoids$^+$聚类算法的 CH 值与运行时间，与原始的 k-medoids 算法进行对比。结果如表 5-7 所示。

从最终聚类结果上来看，表 5-7 统计了 SS-medoids$^+$算法和原始 k-medoids 算法的运行的时间和 CH 值。SS-medoids$^+$算法在聚类的质量上由于原始算法 1%左右，运行效率由于原始算法 50%左右。

从图 5-7 中可以看出，SS-medoids$^+$算法可以将 RNA 二级结构根据结构距离划分到不同的簇内，并产生中心点作为代表结构(红色圆圈部分)说明 SS-medoids$^+$算法

具有聚类分析 RNA 二级结构集合的能力。



(a)PDB_00052　　(b)PDB_00967

(c)CRW_00438　　(d)PDB_01273

图 5-7 RNA 二级结构聚类效果图

表 5-7 SS-medoids^{+}聚类算法与原始 k-medoids 算法比对结果

ID	应用算法	划分类别数	CH	运行时间
1	k-medoids	3	1324.04	6.01s
PDB_00052	SS-medoids^{+}	3	1338.92	3.04s
2	k-medoids	2	4677.71	7.9s
PDB_00967	SS-medoids^{+}	2	4704.02	3.64s
3	k-medoids	2	3237.05	8.64s
CRW_00438	SS-medoids^{+}	2	3240.92	2.99s
4	k-medoids	2	1115.2	6.99s
PDB_01273	SS-medoids^{+}	2	1129.9	2.79s

*表中所有数据为四舍五入的近似值

5.4 本章小结

本章首先对实验环境和数据来源进行了介绍，然后分别对多进程 Rsd-bp 算法和 SS-medoids^{+}算法进行验证。多进程 Rsd-bp 算法实验以 BP 算法和 RBP 算法为对比从距离计算的可行性和计算效率两方面验证算法。SS-medoids^{+}聚类实验首先对实验的评分标准进行了介绍；然后对实验参数进行测试，选择合适参数；接着对数据进行降维，实现聚类的可视化表示；最后将该算法与原始的 k-medoids 算法对比评价分析 SS-medoids^{+}算法性能。

结 论

RNA 二级结构预测是研究 RNA 分子功能特性的关键，聚类分析 RNA 二级结构集合对提高真实 RNA 结构的预测准确性十分重要。为解决已有 RNA 二级结构聚类算法在 RNA 二级结构距离计算上，单纯计算碱基距离忽略 RNA 二级结构间形状间差异，而产生误差性问题，以及在聚类分析算法上，采用传统无监督聚类算法对噪声敏感，依赖于中心点初始化位置的问题。本文提出 RNA 二级结构形状-碱基对距离计算算法和基于 RNA 二级结构的半监督聚类算法，主要研究成果如下。

(1)提出 Rsd-bp 算法，实现了从形状、碱基对两个方面计算 RNA 二级结构间的距离。该算法在区分 RNA 二级结构碱基对差异的基础上可区分茎环位置的差异，为 RNA 二级结构的聚类研究提供依据。同时，提出多进程 Rsd-bp 算法并行优化原始算法，缩短 Rsd-bp 算法对多个 RNA 二级结构距离计算时间，提升了算法效率。

(2)提出 SS-medoids^{+}算法，实现了基于 RNA 二级结构集合的聚类分析。该算法根据 RNA 二级结构数据特点，将半监督聚类思想与 k-medoids 聚类思想融合，并改进原始算法的中心点更新规则，提出可变阈值的搜索方法。提升了聚类算法的稳定性，加快了算法收敛，提高了算法效率。

本文算法根据 RNA 二级结构数据特点，对以往的 RNA 二级结构聚类算法进行分析，并提出了改进方法。虽然在实验验证上其效果有所提高但仍有以下缺陷。

(1)对于 Rsd-bp 算法，Rsd-bp 算法无法计算 RNA 二级结构中假结位置不同引起的距离差异。由于采用玻尔兹曼抽样算法，无法生成有假结的 RNA 二级结构集合，所以 Rsd-bp 算法在设计过程中未考虑假结的存在。

(2)对于 SS-medoids^{+}算法，SS-medoids^{+}算法对于比例系数的选取采取了抽样测试的方式，易产生一定的误差。虽然以估计的方式选取比例系数，可以使计算结果得到提升，但并非最优结果，所以如何选取精确的比例系数仍为进一步研究的目标。

参考文献

[1] Matthew C.60 years ago, Francis Crick changed the logic of biology[J/OL].PLOS Biology,2017,15(9)：e2003243[2019-03-01].https://doi.org/10.1371/journal.pbio.2003243

[2] Kristensen L S,Hansen T B,MT Venø, et al. Circular RNAs in cancer: opportunities and challenges in the field[J]. Oncogene, 2018, 37(5):555-565.

[3] Shujuan Meng,Hecheng Zhou,Ziyang Feng,Zihao Xu,Ying Tang,et al.CircRNA:functions and properties of a novel potential biomarker for cancer[J].Molecular Cancer,2017,16(1):94-99.

[4] Lukasiak P,Antczak M,Ratajczak T, et al.RNAssess—a web server for quality assessment of RNA 3D structures[J].Nucleic Acids Research, 2015, 43(W1):502-506.

[5] 王传铭, 潘珉, 曹槐. RNA 折叠[J]. 自然杂志, 2004, 26(5):249-254.

[6] Tamar Schlick,Anna Marie Pyle. Opportunities and Challenges in RNA Structural Modeling and Design[J]. Biophysical Journal,2017,113(2):225-234.

[7] Behrouzi R,Roh J,Kilburn D,et al.Cooperative Tertiary Interaction Network Guides RNA Folding[J].Cell,2012, 149(2):348-357.

[8] Jr T I, Bustamante C.How RNA folds[J]. Journal of Molecular Biology,1999,293(2):271-281.

[9] Zemora G , Waldsich C .RNA folding in living cells[J].RNA Biology,2010,7(6):634-641.

[10] Tan Z, Fu Y, Sharma G,et al.TurboFold II:RNA structural alignment and secondary structure prediction informed by multiple homologs[J]. Nucleic Acids Research, 2011, 45(20):11570-11581.

[11] Lee H.T.,D Kilburn,S.A. Woodsonet al.Molecular crowding overcomes the destabilizing effects of mutations in a bacterial ribozyme[J].Nucleic Acids Res,2015,43(2):1170-1176.

[12] Nussinov R,Pieczenik G,Daniel J,et al.Algorithms for Loop Matchings[J].SIAM Journal on Applied Mathematics,1978,35(1):68-82.

[13] Zuker M,Sankoff D.RNA secondary structures and their prediction[J]. Bulletin of Mathematical Biology, 1984, 46(4):591-621.

[14] Zuker M.On Finding All Suboptimal Foldings of an RNA Molecule[J].Science,1989, 244(4900):48-52.

[15] Lyngsø RB1,Zuker M,Pedersen CN..Fast evaluation of internal loops in RNA secondary structure prediction[J].Bioinformatics,1999,15(6):440-445.

[16]谭光明,冯圣中,孙凝晖.RNA 二级结构预测中动态规划的优化和有效并行[J].软件学报,2006(7):1501-1509.

[17] Wiese K C, Hendriks A, Deschenes A, et al.Significance of randomness in P-RnaPredict-a parallel evolutionary algorithm for RNA folding[C].//IEEE Congress on Evolutionary, Edinburgh, Scotland,UK,2005:467-474.

[18] Ray S S , Pal S K . RNA Secondary Structure Prediction Using Soft Computing[J].IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2013, 10(1):2-17.

[19] Tsang H H ,Wiese K C.SARNA-Predict:Accuracy Improvement of RNA Secondary Structure Prediction Using Permutation-Based Simulated Annealing[J].IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2010,7(4):727-740.

[20]Liu Q, Ye X, Zhang Y. A Hopfield Neural Network Based Algorithm for RNA Secondary Structure Prediction[C]//First International Multi-Symposiums on Computer and Computational Sciences. Washington,DC,USA,2006:10-16.

[21] Jaeger J A,Turner D H,Zuker M.Improved predictions of secondary structures for RNA[J].Proceedings of the National Academy of Sciences of the United States of America,1989,86(20):7706-7710.

[22] Ding Y ,Lawrence CE.Astatistical sampling algorithm for RNA secondary structure prediction[J].Nucleic Acids Research,2003,31(24):7280-7301.

[23] Ding Y ,Chan C Y,Lawrence C E .Sfold web server for statistical folding and rational design of nucleic acids[J/OL].Nucleic Acids Research,2004,32:135-141[2019-03-01]. https://doi.org/10.1093/nar/gkh449

[24] Ding Y,Chan C Y,Lawrence C E .RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble[J].Rna-a Publication of the Rna Society,2005,11(8):1157-1166.

[25] Ding Y,Chan C Y, Lawrence C E.Clustering of RNA Secondary Structures with Application to Messenger RNAs[J].Journal of Molecular Biology,2006,359(3):554-571.

[26] Agius P, Bennett K P,Zuker M.Comparing RNA secondary structures using a relaxed base-pair score[J].RNA-a Publication of the Rna Society, 2010, 16(5):865-878.

[27] 王常武, 刘小凤, 王宝文等.IC-kmedoids：适用于 RNA 二级结构预测的聚类算法[J]. 生物医学工程学杂志, 2015(1):99-103.

[28] 王秀芹.基于 RBP 的次最优自由能 RNA 二级结构的密度聚类问题研究[D].河北：燕山大学计算机应用技术硕士学位论文,2014:5-9.

[29] Rogers E,Heitsch C.New insights from cluster analysis methods for RNA secondary structure prediction[J].Wiley Interdisciplinary Reviews:RNA,2016,7(3):278-294.

[30] Voß Björn,Robert G , Marc R.Complete probabilistic analysis of RNA shapes[J]. BMC Biology, 2006, 4(1):5-15.

[31] Steffen P,Voss B,Rehmsmeier M,et al.RNAshapes:an integrated RNA analysis package based on abstract shapes[J].Bioinformatics,2006, 22(4):500-503.

[32] Huang J,Backofen R,Voss B.Abstract folding space analysis based on helices[J].RNA,2012, 18(12):2135-2147.

[33] Eddy,S.R. What is a hidden Markov model[J]. Nature Biotechnology,2004.22(10):1315-1316.

[34] Rivas E.,Eddy SR. The Language of RNA:A Formal Grammar That Includes Pseudoknots[J]. Bioinformatics,2000(16):334-340.

[35] Tabei,Yasuo,Kiyoshi Asai. A local multiple alignment method for detection of non-coding RNA sequences. Bioinformatics,2009,25(12):1498-1505.

[36] Sven Siebert,Rolf Backofen. MARNA:A Server for Multiple Alignment of RNAs[C]//Proceedings of the German Conference on Bioinformatics. Berlin ,Germany, 2003:135-140.

[37] Mathews D H, Turner D H. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences[J].Journal of molecular biology,2002,317(2):191-203.

[38] Reuter J,Mathews.RNA Secondary Structure Prediction[J].Bioinformation,2013,9(17):873-883.

[39] Schirmer S,Ponty Y ,Giegerich R.Introduction to RNA secondary structure comparison[M].// Gorodkin J., Ruzzo W. RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods. New Jersey, USA:Humana Press, Totowa, NJ,2014,1097:247-273.

[40] Allali J, Sagot M F. A new distance for high level RNA secondary structure comparison[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2005,2(1):3-14.

[41] Everitt B.Cluster analysis[J].Quality & Quantity, 1980,14(1):75-100.

[42] Roux M. A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms[J].Journal of Classification,2018,35(2):345-366.

[43] Kanungo T,Mount D M, Netanyahu N S, et al.An efficient k-means clustering algorithm:Analysis

and implementation[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002 (7): 881-892.

[44] Hartigan J A,Wong M A.Algorithm AS 136:A K-means clustering algorithm[J].Journal of the Royal Statistical Society, 1979, 28(1):100-108.

[45] 刘雪娟,袁家斌,操凤萍.云计算环境下面向数据分布的 K-means 聚类算法[J].小型微型计算机系统,2017,38(04):712-715.

[46]Alkoffash M S,Alkoffash M S.Automatic Arabic Text Clustering using K-means and K-medoids[J].International Journal of Computer Applications,2012, 51(2):5-8.

[47] Figueiredo M A T , Nowak R D .An EM algorithm for wavelet-based image restoration[J].IEEE Transactions on Image Processing,2003,12(8):906-916..

[48] Pedrycz W.Introducing WIREs Data Mining and Knowledge Discovery[J].Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2011,1(1):1-26.

[49] 周志华.机器学习[M].北京:清华大学出版社,2016：307-309

[50] 李昆仑，曹铮，曹丽苹,等. 半监督聚类的若干新进展[J]. 模式识别与人工智能，2009, 22(5):735-742.

[51] Mai X, Cheng J, Wang S. Research on semi supervised K-means clustering algorithm in data mining[J]. Cluster Computing,2018,21(1):1-8.

[52] Wagstaff K , Cardie C , Rogers S, et al. Constrained K-means Clustering with Background Knowledge[C]// Proceedings of the Eighteenth International Conference on Machine Learning. Weiliansidun, Melbourne, Australia. 2001:77-584.

[53] 管仁初，半监督聚类算法的研究与应用[D].吉林：吉林大学计算机科学与技术博士学位论文, 2010:7-8.

[54] Huang H, Cheng Y, Zhao R. A semi-supervised clustering algorithm based on must-link set[C]// Proceedings of the 4th international conference on Advanced Data Mining and Applications table of contents. Berlin: Springer-Verlag, 2008:492-499.

[55] Davidson Ian, Ravi S.S. Clustering with Constraints: Feasibility Issues and the k-Means Algorithm[J]. SDM,2005, 16(95):1147–1157.

[56] Andronescu M , Bereg V , Hoos H H , et al. RNA STRAND : The RNA Secondary Structure and Statistical Analysis Database[J]. BMC Bioinformatics, 2008, 9(1):340-350

附 录

表中 5-6 中 RNA 的一级序列详细信息

DatabaseID:PDB_00052

GCGGAUUUAgCUCAGuuGGGAGAGCgCCAGAcUgAAgAPcUGGAGgUCcUGUGu
PCGaUCCACAGAAUUCGCACCA

DatabaseID: PDB_00967

GCCGGGGUGGCGGAAUGGGUAGACGCGCAUGACAUCAUGUGCGCAAGCGUG
CGGGUUCAAGUCCCGCCCCCGGCACCAGCCGGGGUGGCGGAAUGGGUAGAC
GCGCAUGACAUCAUGUGCGCAAGCGUGCGGGUUCAAGUCCCGCCCCCGGCA
CCA

DatabaseID: CRW_00438

AAUAGGUUUGGUCCUAGCCUUUCUAUUAGCUCUUAGUAAGAUUACACAUGC
AAGCAUCCCCGUUCCAGUGAGUUCACCCUCUAAAUCACCACGAUCAAAAGG
AACAAGCAUCAAGCACGCAGCAAUGCAGCUCAAAACGCUUAGCCUAGCCAC
ACCCCCACGGGAAACAGCAGUGAUUAACCUUUAGCAAUAAACGAAAGUUUA
ACUAAGCUAUACUAACCCCAGGGUUGGUCAAUUUCGUGCCAGCCACCGCGG
UCACACGAUUAACCCAAGUCAAUAGAACCCGGCGUAAAGAGUGUUUUAGAU
CACCCCCUCCCCAAUAAAGCUAAAACUCACCUGAGUUGUAAAAAACUCCAG
UUGACACAAAAUAGACUACGAAAGUGGCUUUAACAUAUCUGAACACAGAAU
AGCUAAGACCCAAACUGGGAUUAGAUACCCCACUAUGCUUAGCCCUAAACC
UCAACAGUUAAAUCAACAAAACUGCUCGCCAGAACACUACGAGCCACAGCU
UAAAACUCAAAGGACCUGGCGGUGCUUCAUAUCCCUCUAGAGGAGCCUGUU
CUGUAAUCGAUAAACCCCGAUCAACCUCACCACCUCUUGCUCAGCCUAUAU
ACCGCCAUCUUCAGCAAACCCUGAUGAAGGCUACAAAGUAAGCGCAAGUAC
CCACGUAAAGACGUUAGGUCAAGGUGUAGCCCAUGAGGUGGCAAGAAAUGG
GCUACAUUUUCUACCCCAGAAAACUACGAUAGCCCUUAUGAAACUUAAGGG
UCGAAGGUGGAUUUAGCAGUAAACUAAGAGUAGAGUGCUUAGUUGAACAG
GGCCCUGAAGCGCGUACACACCGCCCGUCACCCUCCUCAAGUAUACUUCAA

AGGACAUUUAACUAAAACCCCUACGCAUUUAUAUAGAGGAGACAAGUCGUA
ACAUGGUAAGUGUACUGGAAAGUGCACUUGGACGAAC

DatabaseID: PDB_01273

UGGAGAGUUUGAUCCUGGCUCAGGGUGAACGCUGGCGGCGUGCCUAAGACA
UGCAAGUCGUGCGGGCCGGGGUUUUACUCCGGGUCAGCGGCGGACGGGUGA
GUAACGCGUGGGUACCUACCCGGAAGAGGGGGACAACCCGGGGAAACUCGG
GCUAAUCCCCCAUGUGGACCCGCGUCCAAAGGGCUUUGCCCGCUUCCGGAU
GGGCCCGCGUCCCAUCAGCUAGUUGGUGGGGUAAUGGCCCACCAAGGCGAC
GACGGGUAGCCGGUCUGAGAGGAUGGCCGGCCACAGGGGCACUGAGACACG
GGCCCCACUCCUACGGGAGGCAGCAGUUAGGAAUCUUCCGCAAUGGGCGCA
AGCCUGACGGAGCGACGCCGCUUGGAGGAAGAAGCCCUUCGGGGUGUAAAC
UCCUGACCCGGGACGAAACCCCCACGGGGCUGACGGUACCGGGGUAUAGCG
CCGGCCAACUCCGUGCCAGCAGCCGCGGUAAUACGGAGGGCGCGAGCGUUA
CCCGGAUUCACUGGGCGUAAAGGGCGUGUAGGCGGCCUGGGGCGUCCCAUG
UGAAAGACCACGGCUCAACCGUGGGGGAGCGUGGGAUACGCUCAGGCUAGA
CGGUGGGAGAGGGUGGUGGAAUUCCCGGAGUAGCGGUGAAAUGCGCAGAU
ACCGGGAGGAACGCCGAUGGCGAAGGCAGCCACCUGGUCCACCCGUGACGC
UGAGGCGCGAAAGCGUGGGGAGCAAACCGGAUUAGAUACCCGGGUAGUCCA
CGCCCUAAACGAUGCGCGCUAGGUCUCUGGGUCUCCUGGGGGCCGAAGCUA
ACGCGUUAAGCGCGCCGCCUGGGGAGUACGGCCGCAAGGCUGAAACUCAAA
GGAAUUGACGGGGGCCCGCACAAGCGGUGGAGCAUGUGGUUUAAUUCGAAG
CAACGCGAAGAACCUUACCAGGCCUUGACAUGCUAGGGAAACCCGGGUGAA
AGCCUGGGGUGCCCCGCGAGGGGAGCCCUAGCACAGGUGCUGCAUGGCCGU
CGUCAGCUCGUGCCGUGAGGUGUUGGGUUAAGUCCCGCAACGAGCGCAACC
CCCGCCGUUAGUUGCCAGCGGUUCGGCCGGGCACUCUAACGGGACUGCCCG
CGAAGCGGGAGGAAGGAGGGGACGACGUCUGGUCAGCAUGGCCCUUACGGC
CUGGGCGACACACGUGCUACAAUGCCCACUACAAAGCGAUGCCACCCGGCA
ACGGGGAGCUAAUCGCAAAAAGGUGGGCCCAGUUCGGAUUGGGGUCUGCAA
CCCGACCCCAUGAAGCCGGAAUCGCUAGUAAUCGCGGAUCAGCCAUGCCGC

GGUGAAUACGUUCCCGGGCCUUGUACACACCGCCCGUCACGCCAUGGGAGC
GGGCUCUACCCGAAGUCGCCGGGAGCCAGGCGCCGAGGGUAGGGCCCGUGA
CUGGGGCGAAGUCGUAACAAGGUAGCUGUACCGGAAGGUGCGGCUGGAUCA
CUUUCUCGGGUCCCGAAU

致 谢

时光如水，岁月如歌。转眼间，三年的研究生生活即将结束，站在毕业的门槛上，回首往昔，学习和求知的过程还在脑海浮现，和同学之间的同窗之情还在心间萦绕。燕山大学以其严谨的科研氛围，开拓性的知识探索方法帮我在科研的路上前行；以其海纳百川的胸怀和充实热情的校园生活助我成人。值此毕业论文完成之际，我谨向所有关心、爱护、帮助我的人们表示最诚挚的感谢与最美好的祝愿。

感谢我的老师王常武教授在学习和研究过程中给予我的指导和关怀。本论文是在王老师的悉心指导下完成的，从论文的选题、实验构思、实验操作到论文的撰写等各个方面，王老师言传身教，付出了大量的心血。在研一的时候，导师让我们阅读了大量相关文献，为我们研究的内容打下了坚实的基础。同时每周一次的例会不仅让我们对自己的学习有个硬性规定，也使得与同门师兄弟们的思想进行了交流。

感谢柯铁军老师，王亮老师，司亚利老师在学术和生活上的指导与帮助，感谢实验室同门马延龙、袁芹等人在科研中的经验的交流和思维的启发，在遭遇学术瓶颈是的帮助与探讨。感谢舍友袁梅，赵旋在日常生活中的照顾，在我生病时带给我的温暖与照顾，在学习中资料的分享与思路的探讨。我相信和你们在一起的日子将会是我一生中最美好的回忆。

感谢每一个关心我、帮助我的人，谢谢你们，祝愿大家永远快乐健康平安！

感谢在百忙之中审阅论文和参加答辩的各位专家、教授，请给予批评指正！