L-MinHash：一种蛋白质同源性搜索的高效算法

答 辩 人：项溢馨

指导老师：王和兴 老师

2022 年 6 月 9 日

东北大学秦皇岛分校计算机科学与技术专业学士学位论文答辩

内容提要

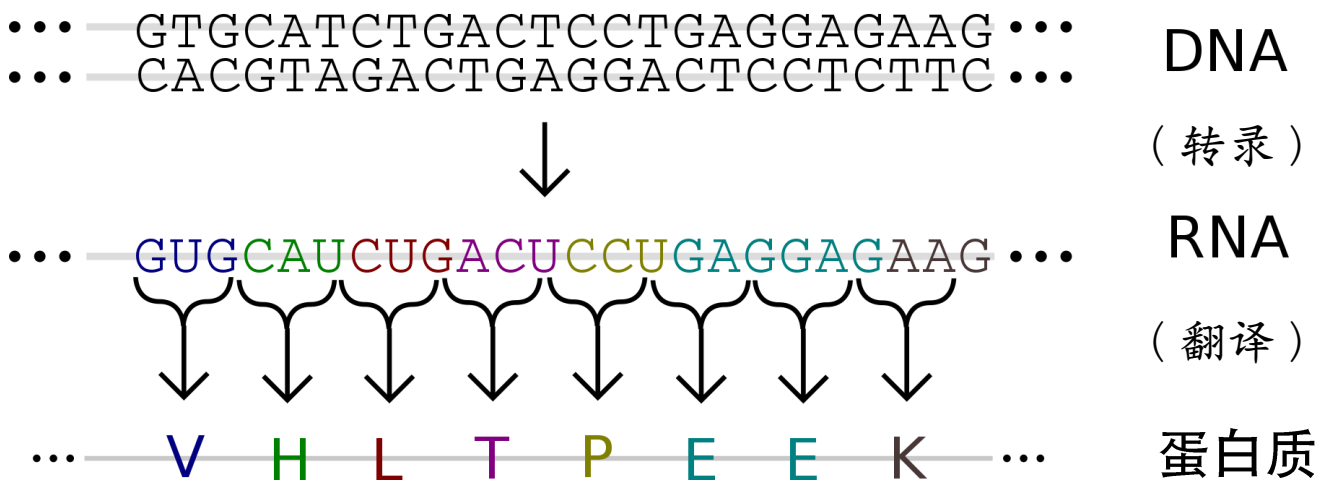
内容提要

选题背景与研究意义

分子生物学

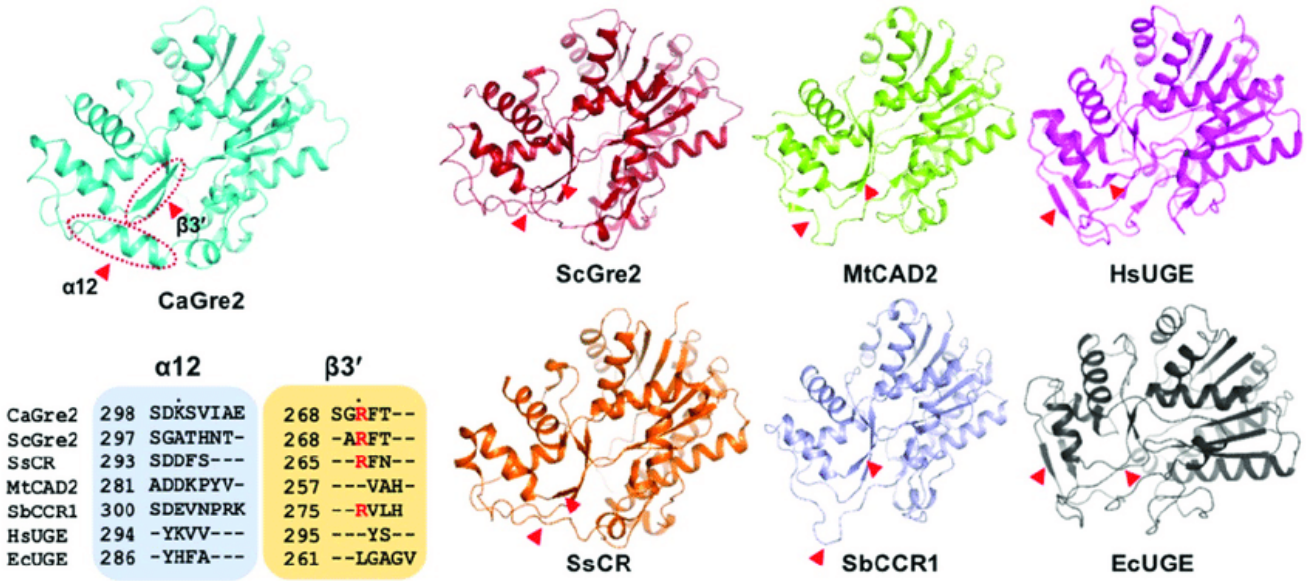
- DNA：由A、C、G、T四种碱基
- RNA：由A、C、G、U四种碱基
- 蛋白质：由 20 种氨基酸

组成的分子序列
（字符串）



选题背景与研究意义

同源蛋白质



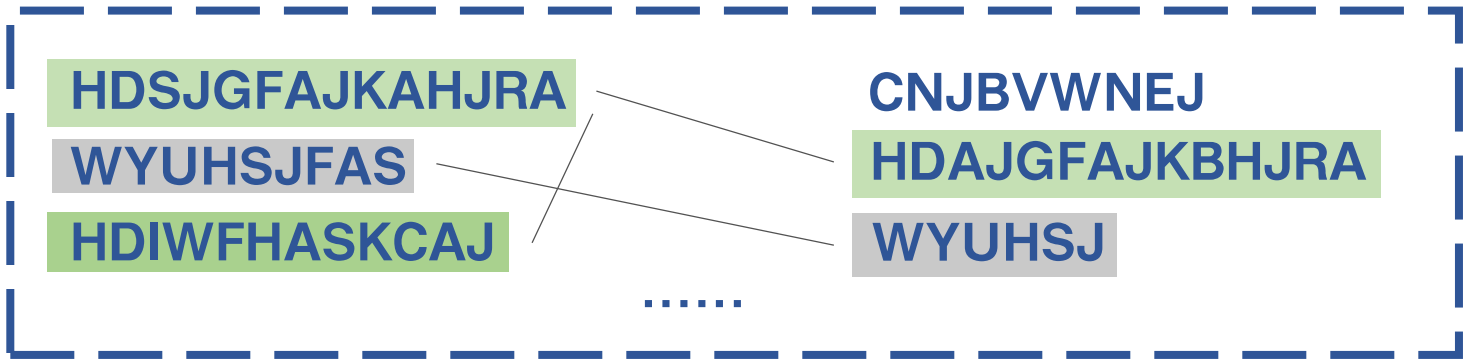
	α12		β3′
CaGre2	298	SDKSVIAE	268 SGRFT--
ScGre2	297	SGATHNT-	268 -ARFT--
SsCR	293	SDDFS---	265 --RFN--
MtCAD2	281	ADDKPYV-	257 ---VAH-
SbCCR1	300	SDEVNPRK	275 --RVLH
HsUGE	294	-YKVV---	295 ---YS--
EcUGE	286	-YHFA---	261 --LGAGV

- 由一个共同祖先演化而来的多个结构间所具有的共性特征
- 高度相似性意味两个序列极有可能拥有共同祖先
- 蛋白质的氨基酸序列决定其三维结构，从而决定其功能特性
  ->进一步分析，应用（制药等）

内容提要

蛋白质同源性搜索问题

蛋白质同源性搜索问题



编辑相似度$E_s(a,b)$

- 编辑相似度的定义基于编辑距离
- 编辑距离指两个字符串之间,通过增删改的编辑操作，由一个转换成另一个所需的最少操作次数

增	删	改
AB	ABB	AC
ABB	AB	AB

编辑相异度

$$E_d(a,b) = E_d(b,a) = \frac{lev(a,b)}{max(|a|,\ |b|)}$$ 编辑距离

$$E_s(a,b) = 1 - E_d(a,b)$$

蛋白质同源性搜索问题

动态规划求解编辑距离 $O(l_a l_b)$

		A	B	B	C
	0	1	2	3	4
A	1	0	1	2	3
C	2	1	1	2	3
B	3	2	2	1	2

$$\mathrm{lev}(a,b)=\begin{cases}|a| & if\ |b|=0,\\ |b| & if\ |a|=0,\\ lev(\mathrm{tail(a)},\mathrm{tail(b)}) & if\ a[0]=b[0],\\ 1+min\begin{cases}lev(\mathrm{tail(a)},\mathrm{b})\\ lev(\mathrm{a},\mathrm{tail(b)})\\ lev(\mathrm{tail(a)},\mathrm{tail(b)})\end{cases} & otherwise\end{cases}$$

蛋白质同源性搜索问题的时间复杂度 $O(n^2 l^2)$

n：蛋白质序列数量 8×10^6 $3\times10^{18}\approx3\times10^{10}$秒

l：蛋白质序列平均长度 200 ≈1000年

蛋白质同源性搜索问题

研究现状及挑战

- 序列比对算法： Smith-Waterman算法和Needleman-Wunsch算法及其优化
- 启发式策略加速：FASTA和BLAST软件
- 哈希算法避开序列直接比较：Mash、Mashmap和Mhap
- 挑战：超长的蛋白质序列和蛋白质序列规模庞大
- 目标：在可接受时间范围内快速搜索出高相似度的序列对

内容提要

蛋白质同源性搜索的高效算法L-MinHash

设计思路

- 对蛋白质序列进行聚类，在更小的类中计算编辑相似度

HDSJGFAJKAHJRA
HDAJGFAJKBHJRA
HDIWFHASKCAJ
……
CNJBVWNEJ
WYUHSJ
WYUHSJFAS

长度	相似度	概率
1	100%	1/20
2	50%	1/160

如果将 8×10^6 条序列分成10000簇

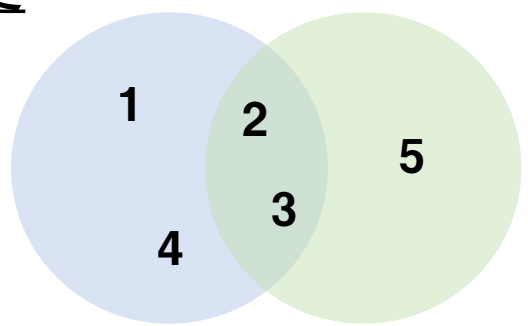
在每个簇中分别求解　时间复杂度从 3×10^{18} 变成 3×10^{14}

- 何种距离度量　~~编辑距离~~　$O(l^2)$
- 何种聚类算法　~~k-means聚类~~　$O(nkt)$

蛋白质同源性搜索的高效算法L-MinHash

距离度量：Jaccard相似度

$$J_s(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

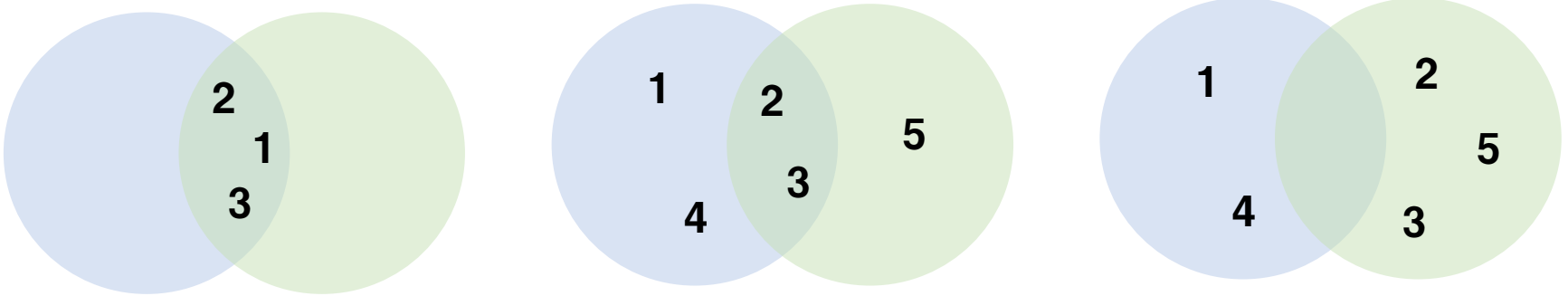


- Jaccard相似度是编辑相似度的上界（论文中已证明）
- Jaccard相似度不超过阈值，则编辑相似度一定不高于阈值，两个序列肯定不相似
- 计算的时间复杂度$O(n+m)$

蛋白质同源性搜索的高效算法L-MinHash

Jaccard相似度估计的近似算法MinHash

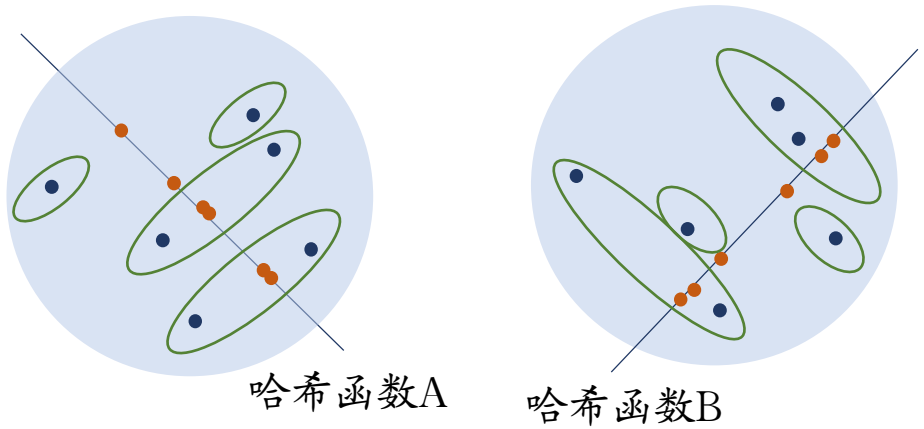
$$J_s(A,B) = \frac{|A \cap B|}{|A \cup B|}$$



- 交集中的数命中最小值则MinHash相等
- $\Pr[h_{min}(A) = h_{min}(B)] = J_s(A,B)$
- 重复多次hash即可估计交集大小

蛋白质同源性搜索的高效算法L-MinHash

聚类算法：基于Jaccard相似度的局部敏感哈希（LSH）

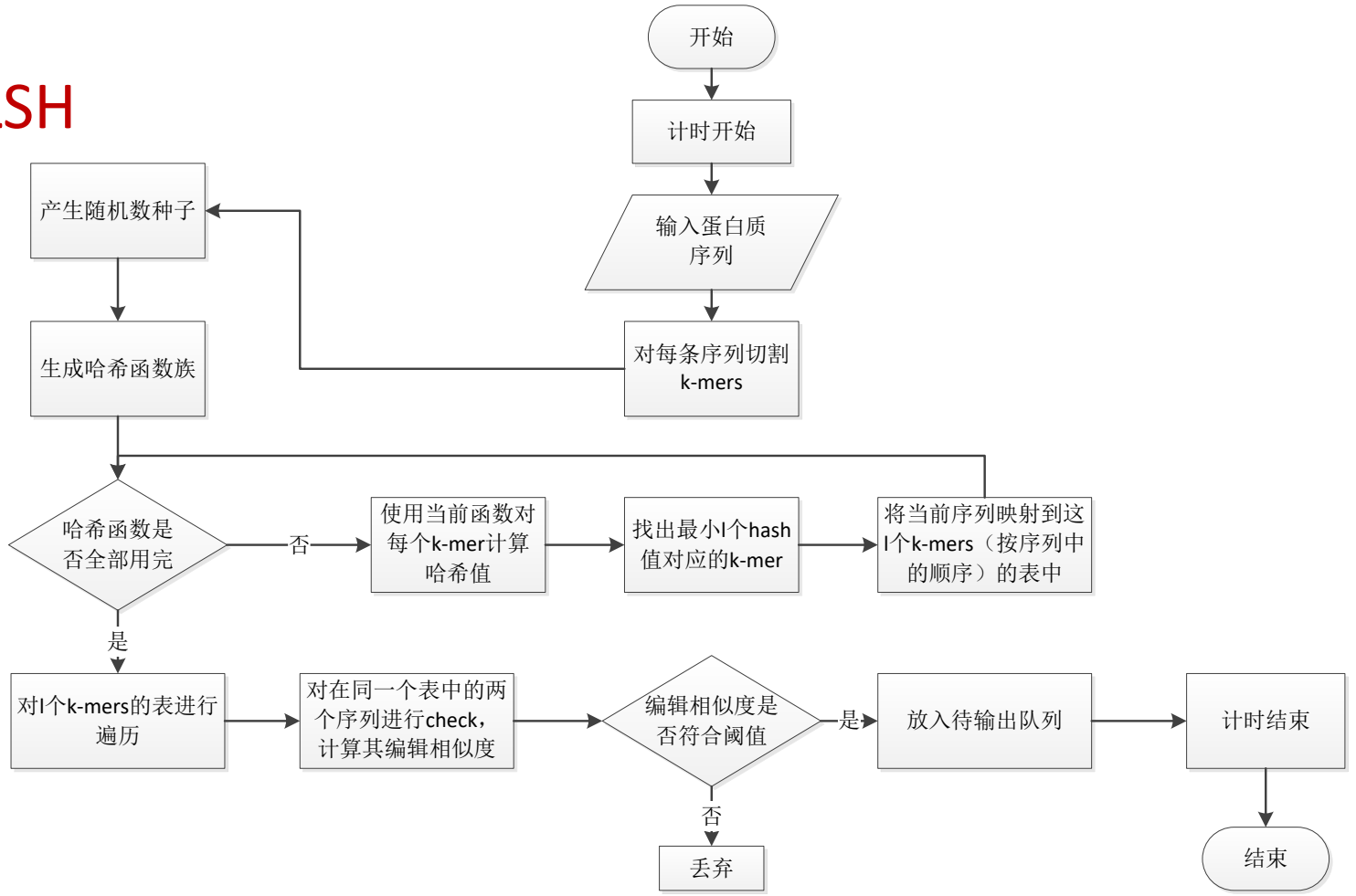


- 效率高，步骤简洁，误差大
- 线性的时间复杂度
- 增加哈希函数个数可以迅速降低误差率

蛋白质同源性搜索的高效算法L-MinHash

算法框架

L-MinHash = MinHash + LSH



蛋白质同源性搜索的高效算法L-MinHash

算法伪代码（部分）

算法 4.1　k-mer 分割算法

```
输入：长度为 L 的蛋白质序列 S，k-mer 的长度 k
输出：输入序列 S 的全部 k-mer 加权集合 V
1: V ← [ ]                          //初始化集合
2: for i = 1 to L-k+1               //每条序列拥有的 k-mer 数量为 L-k+1
3:    s ← S[i..i+k-1]
4:    V ← V ∪ [s, H[s]]
5:    H[s] = H[s]+1
6: end for
```

算法 4.2　Order Min Hash 算法

```
输入：n 组蛋白质序列 S[]，k-mer 的长度 k，散列函数模数 p，全域散列模数 pp
输出：聚类后存在同源性可能的序列对集合 V
1: V ← [ ]
2: let A[1..L] be a new array          //A 存放散列函数的基数
3: for i=1 to L
4:    A[i] = RANDOM(1, p-1)            //生成 L 组散列函数
5: end for
6: for ii = 1 to L
7:    for i = 1 to n
8:       m ← |k_mer[i]|                //m 是第 i 个序列 k-mer 的数量
9:       for j = 1 to m
10:         x ← k_mer[i][j]
11:         val = 1                    //val 用来计算 k-mer 的散列值
12:         for jj = 1 to k
13:            val = ( val * A[ii] + x[jj] ) % p
14:         end for
15:         hash[j] = val % pp
16:      end for
17:      select hash[1..m] corresponding to the smallest l values
         //将最小的 l 个值放置在 hash 序列最左边，l 个值内部保持原有序列排序
18:      for j = 1 to l
19:         v ← v ∪ hash[j]
20:      end for
21:      set[v] ← set[v] ∪ i      //对包含 l 个值和相对位置信息的列表做一个散列
22:   end for
23:   for i = 1 to |set[ ]|
24:      for j = 1 to |set[i]|
25:         for jj = j + 1 to |set[i]|
26:            V ← V ∪ ( set[i][j] , set[i][jj] )      //散列相同的两个序列可能同源
27:         end for
28:      end for
29:   end for
30: end for
31: return V
```

算法 4.3　动态规划求解编辑相似度

```
输入：长度分别为 l1, l2 的蛋白质序列 S1 和 S2
输出：输入序列 S1 和 S2 的编辑相似度 s
1: let dp[0..l1][0..l2] be a new array
2: for i = 0 to l1-1
3:    dp[i][0] = i
4: end for
5: for i = 0 to l2-1
6:    dp[0][i] = i
7: end for
8: for i = 1 to l1
9:    for j = 1 to l2
10:      if S1[i] = S2[j]
11:         dp[i][j] = dp[i-1][j-1]
12:      else dp[i][j] = 1 + min{ dp[i-1][j], dp[i][j-1], dp[i-1][j-1] }
13:   end for
14: end for
15: s = 1 - dp[l1][l2] / max(l1, l2)
15: return s
```

内容提要

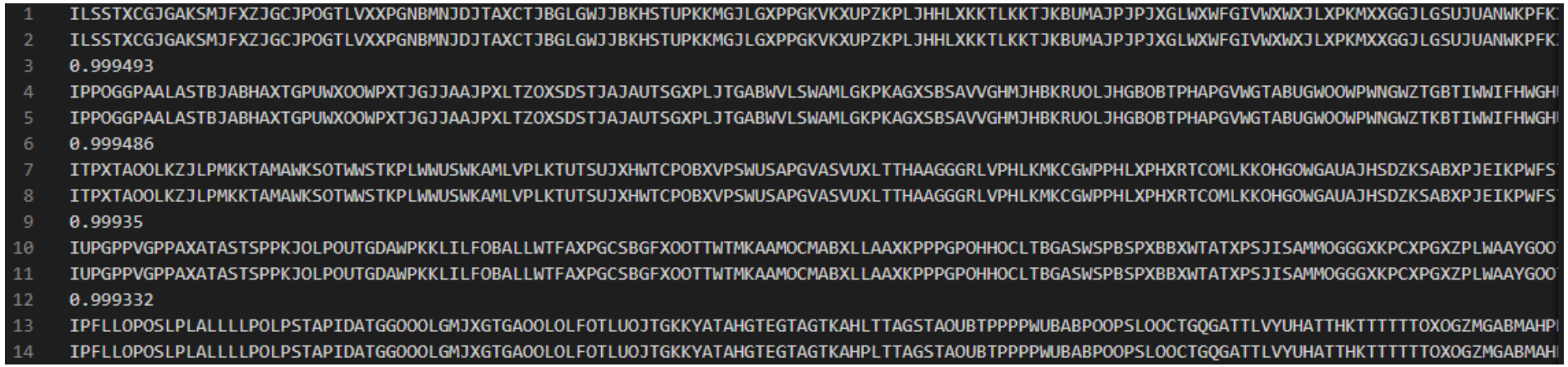
实验结果

模型的效率与准确性的对比

- 初步模型相较于现有模型（Order Min Hash，2019）提速约30%
- 在一些适当参数下，初步模型能达到较高的准确率

初步模型在真实数据集上的效果

- 在将 8×10^6 条序列中共找到992821对相似度高于0.7的序列对



内容提要

总结与期望

- 创新点和主要贡献
- 独立证明了序列间的Jaccard相似度是编辑相似度的上界（算法应用的理论基础）
- 利用最小哈希和局部敏感哈希设计了L-MinHash算法（最小哈希实现了Jaccard相似度的快速估计，局部敏感哈希基于相似度估计值对全部序列进行了聚类）
- 复现了已有的模型Order Min Hash，并与之进行比较

- 实验结果：分别在模拟数据和真实数据上对不同算法的性能进行了测试，实验结果验证了新算法具有出色的性能优势，值得进行深入研究。
- 不足：系统框架较粗糙，算法性能仍有待改进
- 目标：下一步继续完善算法，进行实验，对论文进行细致修改后投稿。并对系统进行封装，使之可以真正应用于生物学研究。

谢　谢

请各位老师批评指正