序列比对及BLAST应用

目录

- 序列比对简介
  - 序列比对的定义和意义
  - 序列比对的要素
- 序列比对算法
  - 两序列比对
  - 多序列比对
- BLAST软件的使用
  - 软件的算法及功能简介
  - 操作规范

序列比对简介

- 序列比对的定义和意义
- 序列比对的要素

序列比对的定义和意义

- 序列比对的定义
  - 将两条或多条DNA、RNA或蛋白质序列排列在一起，标出其中相似的位点。
- 序列比对的意义
  - 判断两条序列相似性的基本方法
  - 推断序列间的功能、结构和进化关系

序列比对的要素

- 评判序列差异的规则——打分规则
  - 序列匹配的打分矩阵
  - 序列空位的罚分规则
- 实现序列比对的算法
- 评价序列比对结果统计显著性的方法

氨基酸序列的打分原则

Point Accepted Mutation Matrix

PAM250

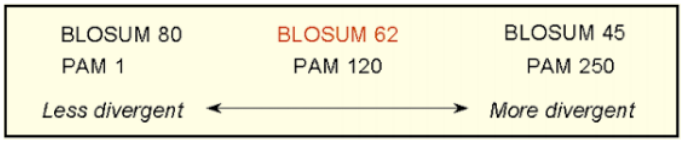
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	4															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-4	-4	-4	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	-1	-1	-3	0	-1	-1	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	3				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-2	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

BLOcks SUbstitution Matrix

BLOSUM62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

尽管两个矩阵推导的过程不同，



Matrices used in BLAST Source: NCBI

核苷酸序列的打分规则

- Hamming 距离:

$$d_H(a,b) = \begin{cases} 0, & if\ a = b, \\ 1, & otherwise. \end{cases}$$

  - BLASTN 打分矩阵:

$$\begin{pmatrix} Similarity & Reward & Penalty \\ 99\%(PAM1) & 1 & -3 \\ 95\%(PAM5) & 1 & -2 \\ 75\%(PAM30) & 1 & -1 \end{pmatrix}$$

States DJ, Gish W, and Altschul SF (1991) Improved sensitivity of nucleotide acid database searches using application-specific scoring matrices. METHODS: A companion to Methods in Enzymology 3:66-70.

序列比对算法

- 两序列比对
- 多序列比对

序列比对算法

两序列比对

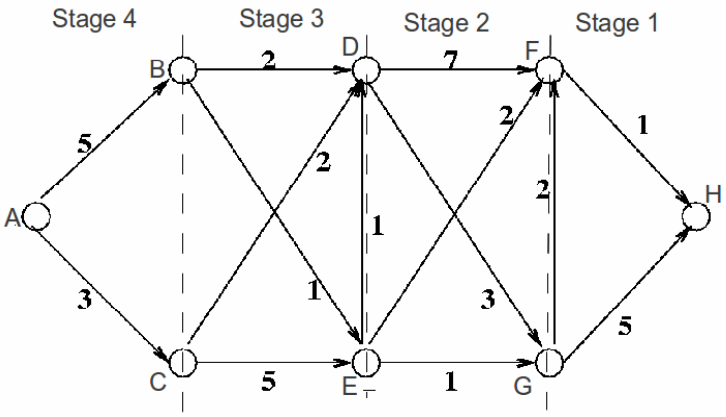
- 动态规划算法
  - Needleman-Wunsch 全局序列比对算法(1970)
  - Smith-Waterman 局部序列比对算法(1981)
- 启发式算法
  - BLAST 以种子序列为基础的局部比对算法 (Altschul *et al.*, 1990)

多序列比对

- 渐进式算法
  - Clustal V, W, X, ω 等算法，依照两序列比对的结果，构建指导树，再进行多序列比对。 (Higgins and Thompson, 1992, 1994, 2011)

动态规划算法

- 动态规划算法
  - 将问题划分成相互独立的子问题
  - 每一个子问题都找到最优解
  - 记录下每一步的选择并回溯所有的选择



全局比对 vs. 局部比对

全局比对

- 对序列整体进行的比对
- 适于相似度较高的序列

Needleman-Wunsch Algorithm:
Default Initialization:

$$F(0,0) = 0.$$

Recurrence relation:

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

Final score is $F(m,n)$.

局部比对

- 对子序列进行的比对
- 适于搜寻保守的子序列（功能域、保守域）

Smith-Waterman Algorithm:
Default Initialization:

$$F(0,0) = 0.$$

Recurrence relation:

$$F(i,j) = \max \begin{cases} 0, \\ F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

Final score is F_{max}.

全局比对 vs. 局部比对

全局比对

局部比对

假设：匹配得1分, 不匹配得 -3分, 空位罚分-2分。

		A	C	G	T	C
	0	-2	-4	-6	-8	-10
A	-2	1	-1	-3	-5	-7
G	-4	-1	-2	0	-2	-4
T	-6	-3	-4	-2	1	-1
C	-8	-5	-2	-4	-1	2
A	-10	-7	-4	-5	-3	0
G	-12	-9	-6	-3	-5	-2

ACGTC- -
A - GTCAG

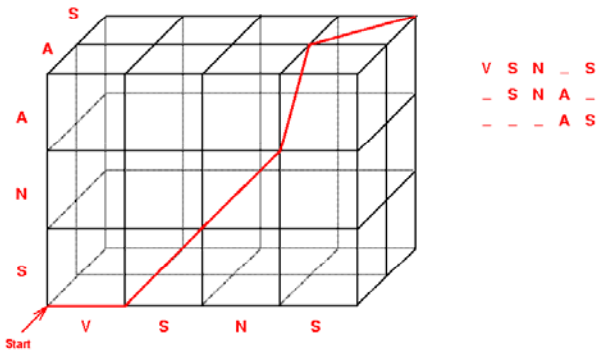
		A	C	G	T	C
	0	0	0	0	0	0
A	0	1	0	0	0	0
G	0	0	0	1	0	0
T	0	0	0	0	2	0
C	0	0	1	0	0	3
A	0	1	0	0	0	0
G	0	0	0	1	0	0

ACGTC. .
A. GTCAG

多序列比对算法

高维动态规划

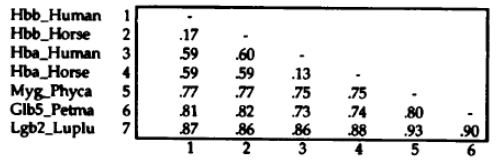
- 时间复杂度: $O(2^N \bar{L}^N)$
- 步骤:

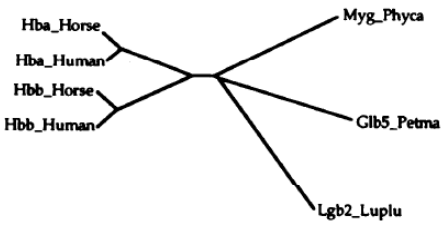


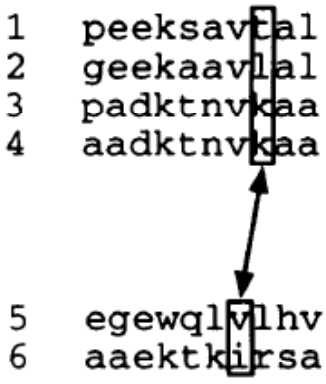
渐进式序列比对

- 时间复杂度: $O((\bar{L}N)^2)$
- 步骤:
  - 对所有N(N-1)/2 对序列进行两序列比对
  - 构建指导树
  - 渐进的比对在指导树上距离较近的序列

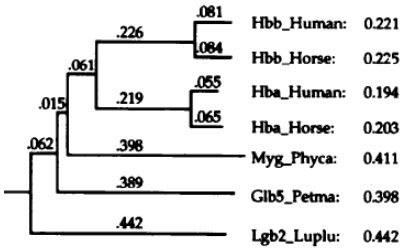
以Clustal W的计算步骤为例

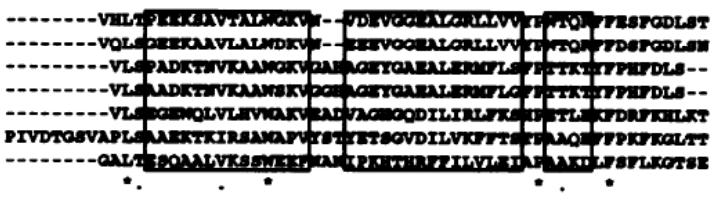






Profile





BLAST软件的使用

- 软件的算法及功能简介
- 操作规范

软件的算法及功能简介

- Basic Local Alignment Search Tool
  - 高效的局部序列比对算法
  - 用已知序列建立数据库，再输入查询序列与数据库中的序列进行比对
  - 序列比对结果有助于对查询序列的结构及功能进行推断
  - 输入文件必须是fasta格式（如下图所示）

```
>r1.1
CTCAAGAAGTGAAGCCATCAGAGATGCAATAAGGACGTATATAACCGAGTACAGGTGGCTTGAGAGCGAAAAAGGGGAGATTGTGGGCGTTCTTGTGGTG
>r2.1
CTTGACGAGATAATCGACAGACTCGCAAGCTTGGGGCTGGCGGAGAAGGAAGGGGACGTTTACAGAATTGATTTGGCGGAACTTGGCTACAGCAAGCTTC
```

http://blast.ncbi.nlm.nih.gov/Blast.cgi

BLAST的算法简介

- 启发式序列比对算法
  - 将输入序列拆分成种子序列(氨基酸种子序列长度为3，核苷酸种子序列长度为11。)
  - 在数据库中的所有序列上搜索所有可能匹配的种子序列位点
  - 以上述位点为基础向两侧延伸用动态规划算法进行比对直至得到最大得分

BLAST比对结果的评价

- Expectation-value：在完全随机的条件下，有多少条序列的得分超过用户规定的得分的期望

E-value:

$$E = Kmne^{-\lambda S}$$

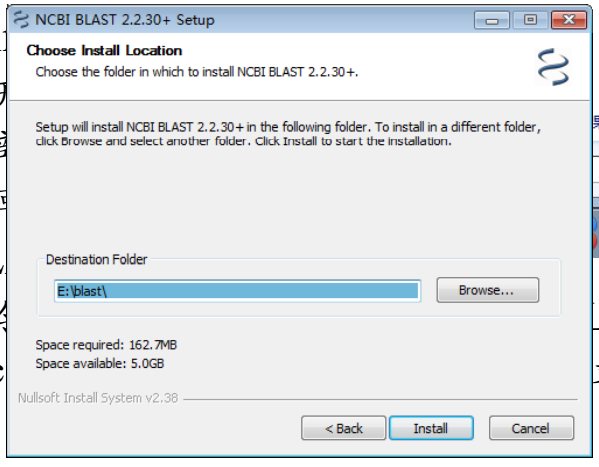
- 上式中，K和λ均为常数，S为两序列比对的得分，m为数据库序列的长度，n为查询序列的长度。
- E-value越低则显著性越高

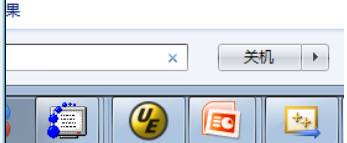
软件的算法及功能简介

- 主要的比对功能
  - blastn:查询序列与数据库序列均为核苷酸序列
  - blastp:查询序列与数据库序列均为氨基酸序列
  - blastx:查询序列为核苷酸序列，数据库序列是氨基酸序列，比对时将核苷酸序列翻译成氨基酸序列再进行比对
  - tblastn: 输入序列为氨基酸序列，数据库序列是核苷酸序列，比对时将数据库序列翻译成氨基酸序列再进行比对
  - tblastx: 输入序列和数据库中的序列均为核苷酸序列，但比对时将上述序列都翻译成氨基酸序列，再进行比对

操作规范

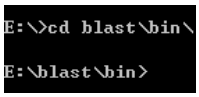
1. 使用BLAST的准备工作
   1. 安装BLAST软件
      1. 双击ncbi-blast-2.2.30+-win32.exe
      2. 在弹出的窗口中输入安装路径
   2. 进入Wi
      1. 打开
      2. 在底
      3. 单击
   3. 进入BL
      1. 在命
      2. 输入c





LAST的硬盘

文件夹



操作规范

2. 构建序列比对数据库
   - 输入makeblastdb.exe –in [存有数据库序列的文件(fasta 格式)] –dbtype [选项 nucl/prot/string] –out [数据库名称]
   - 例如：

```
E:\blast\bin>makeblastdb.exe -in E:\example\nucleotide_example.fa -dbtype nucl -
out nucl_example


Building a new DB, current time: 12/18/2015 02:59:07
New DB name:   nucl_example
New DB title:  E:\example\nucleotide_example.fa
Sequence type: Nucleotide
Keep Linkouts: T
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 1 sequences in 0.0301377 seconds.

E:\blast\bin>
```

操作规范

3. 应用BLAST进行序列比对
   1. 选择合适的BLAST命令(blastn\blastp\blastx\tblastn\tblastx)
   2. 输入 blastn\blastp\blastx\tblastn\tblastx.exe -query [存有查询序列的文件(fasta 格式)] -db [数据库名称] -evalue [设定E-value的阈值] -outfmt [输出文件格式] –out [输出文件]
   3. 例如：

```
E:\blast\bin>blastn.exe -query E:\example\query_nucl.fa -db nucl_example -evalue
 0.1 -outfmt 7 -out E:\example\nucl_example.dat

E:\blast\bin>
```

操作规范

4. 输出结果讲解

1. 输入的查询序列：

```
>query sequence
GAAAGAGCGAGGAAGAGCTTGACGAATTCTTCAAAAGCGGAGTTGAAATCAGACCAAAATGGGATTGAAA
GTTGCATTTACATCGGCAGAATTGGTCTCGTCGGAAGGCATGTTGAAATCAGACCAAAATGGGATTGAAA
GCGGTTCTCTTACGTACTCATGCGAGAAGTGAGACTCGCGTTGGTTGAAATCAGACCAAAATGGGATTGA
AAGAGCAAGTCGTGAAACTGAGCAGTCAAAACAGATCGTTAGTTGAAATCAGACCAAAATGGGATTGAAA
GTTTTCCCATACAATTACGACTTCGCCGGAAAAAAAGTTGAAATCAGACCAAAATGGGATTGAAAGAGCG
AGTTCGACCACGTCGTAGGTCTGCTGTCGGCAAGTTGAAATCAGACCAAAATGGGATTGAAAGTGTTTGA
AGTAGTTGAATACACCGTTGTGCTGTTTGTTGTTGAAATCAGACCAAAATGGGATTGAAAGAGAGGGAGT
ATTAGGGCCATACTGGCCGGAGTTGTGGTTGTTGAAATCAGACCAAAATGGGATTGAAAGATTCCAAATT
GCGGAAAAAGATTCGAGGGCAGTTACTTCCCGTTGAAATCAGACCAAAATGGGATTGAAAGACGTCGTTT
ATTGCCGTAACGCTAACAC
```

2. 数据库序列：

```
>database sequence
GAAAATGGTTTAATCGGAAATTGAGTAGGAGGATAAAAGTCGCATGCTATTATAAATGAGATGCACTTTC
GACACCTCGCGGAAGTATATAAATGAAAGAAGCCCTCAGAAAACTTTAAATTGGAAATAGAGGGAAAATT
ACTGATGGTTGAAATCAGACCAAAATGGGATTGAAAGAGCCTTTCAGCCCTAGTGTGAGTGTCAGGTTTA
CTCTTGTTGAAATCAGACCAAAATGGGATTGAAAGGTTTGTTAAGGGCTTTGATTTGCTCCTCGGTGGCT
CTGGTTGAAATCAGACCAAAATGGGATTGAAAGTAAAGCAGTTCACCCCTGTTACTGGTTTAACTGCCTT
GTTGAAATCAGACCAAAATGGGATTGAAAGGTATTTGAATCAATGAAAAGAAATCTTACCTCGTCGTTGA
AATCAGACCAAAATGGGATTGAAAGAGTCTTCTGGATGGGTCACAAGGGAGACATCGAGGCGTTGAAATC
AGACCAAAATGGGATTGAAAGTCAGCAAGGTTACGTCGGAGATCCTCGAAGAGGGTATCAGTTGAAATCA
GACCAAAATGGGATTGAAAGCGAGGATTGCTGCCAAAGAGAGCGCCTCGTTCTTCGGTTGAAATCAGACC
AAAATGGGATTGAAAGAAAGTGAACATGCTTAAAGAAATGCTGACAGAAATTGAGTTGAAATCAGACCAA
AATGGGATTGAAAGAGCGAGGAAGAGCTTGACGAATTCTTCAAAAGCGGAGTTGAAATCAGACCAAAATG
GGATTGAAAGTTGCATTTACATCGGCAGAATTGGTCTCGTCGGAAGGCATGTTGAAATCAGACCAAAATG
```

操作规范

3. 在以下输入的情形下

```
E:\blast\bin>blastn.exe -query E:\example\query_nucl.fa -db nucl_example -evalue
 1 -outfmt 7 -out E:\example\nucl_example.dat

E:\blast\bin>_
```

输出结果为：

```
# BLASTN 2.2.30+
# Query: query sequence
# Database: nucl_example
# Fields: query id, subject id, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit score
# 181 hits found
query database  100.00  649 0 0 1 649 710 1358  0.0  1199
query database  77.29 251 30  20  41  281 2334  2567  2e-031    122
query database  73.98 392 59  33  42  413 482 850 2e-030    119
query database  74.77 321 51  24  317 626 148 449 9e-030    117
query database  71.31 603 122 43  42  621 416 990 5e-027    108
query database  74.06 320 49  29  42  346 2403  2703  9e-025    100
query database  73.99 323 49  26  181 481 681 990 3e-024  99.0
query database  75.20 254 39  22  41  283 1642  1882  3e-024  99.0
query database  73.90 318 49  30  317 621 284 580 1e-023  97.1
query database  78.03 173 24  14  184 346 1574  1742  1e-023  97.1
query database  77.46 173 29  10  112 282 2943  3107  4e-023  95.3
query database  71.95 467 84  37  112 554 1440  1883  2e-022  93.5
query database  71.64 536 90  42  42  551 216 715 5e-022  91.6
```

操作规范

- 注意事项

1. 安装BLAST软件、存放查询序列和数据库序列文件时，文件路径应尽量简单，且不要有空格和汉字字符。
2. 为方便大家完成大作业，请留有5G左右的硬盘空间。当然也可以直接使用已经编译好的BLAST程序免去安装步骤。
3. 请将所有的查询序列存入一个文件，所有的数据库序列也存入一个文件，上述文件的格式均必须是fasta格式。
4. 关于-evalue参数的取值：E-value的阈值必须大于1，严格的阈值取1.0e-5，不严格的情况取1。
5. 关于-outfmt参数的取值：输出格式有12种，常用的是7。所有输出形式详见右图。
6. 如果某条查询序列没有相应的比对结果，则说明该查询序列的比对结果均无法满足E-value的阈值。
7. 需要了解更多参数要求的同学可以在调用可执行程序时，输入-help查阅该程序的全部参数选项。

BLAST 安装程序下载地址：
ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.30/
ftp://162.105.160.5/pub/software/BLAST/setup
已编译好的版本下载地址：
ftp://162.105.160.5/pub/software/BLAST/program
相关示例下载地址：
ftp://162.105.160.5/pub/software/BLAST/example

```
-outfmt <String>
  alignment view options:
    0 = pairwise,
    1 = query-anchored showing identities,
    2 = query-anchored no identities,
    3 = flat query-anchored, show identities,
    4 = flat query-anchored, no identities,
    5 = XML Blast output,
    6 = tabular,
    7 = tabular with comment lines,
    8 = Text ASN.1,
    9 = Binary ASN.1,
   10 = Comma-separated values,
   11 = BLAST archive format (ASN.1)
   12 = JSON Seqalign output
```

北京大学定量生物学中心
Center for Quantitative Biology
Center for Theoretical Biology

谢谢！