Course organization

- **Introduction (Week 1-2)**
  - Course introduction
  - A brief introduction to molecular biology
  - A brief introduction to sequence comparison
- **Part I: Algorithms for Sequence Analysis (Week 3 - 8)**
  - Chapter 1-3, Models and theories
    - » Probability theory and Statistics (Week 3)
    - » Algorithm complexity analysis (Week 4)
    - » Classic algorithms (Week 5)
  - Chapter 4. Sequence alignment (week 6)
  - Chapter 5. Hidden Markov Models (week 7）
  - Chapter 6. Multiple sequence alignment (week 8)
- **Part II: Algorithms for Network Biology (Week 9 - 16)**
  - Chapter 7. Omics landscape (week 9)
  - Chapter 8. Microarrays, Clustering and Classification (week 10)
  - Chapter 9. Computational Interpretation of Proteomics (week 11)
  - Chapter 10. Network and Pathways (week 12,13)
  - Chapter 11. Introduction to Bayesian Analysis (week 14,15)
  - Chapter 12. Bayesian networks (week 16)

Chapter 4: Blast

Chaochun Wei

Spring 2018

Contents

- Reading materials
- Introduction to BLAST
- Inside BLAST
  - Algorithm
  - Karlin-Altschul Statistics

Reading materials

Karlin, S, and SF Altschul (1990), "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes", PNAS 87:2264-68
Altschul, SF, Gish, W, Miller, W, Myers, E, Lipman DJ (1990), "Basic Local Alignment Search Tool", J. Mol. Biol. 215:403-410

Supporting materials

Altschul, SF(1991), "Amino Acid substitution matrices from an information theoretic perspective", J. Mol. Biol. 219:555-65
Altschul, SF (1993), "A protein alignment scoring system sensitive at all evolution distances", J. Mol. Biol. 36:290-330
Altschul, SF, and W. Gish (1996), "Local alignment statistics", Methods Enzymol. 266:460-80
Altschul, SF, Bundschuh, R, Olsen, R, and T Hwa (2001). "The estimation of statistical parameters for local alignment score distributions", Nucl. Acids. Res. 29:351-61
Karlin, S, and SF Altschul (1993). "Applications and statistics for multiple high-scoring segments in molecular sequences". PNAS, 90:2264-68
Pearson, WR (1998), "Empirical statistical estimates for sequence similarity searches", J. Mol. Biol. 276:71-84.

Introduction to BLAST

- What is BLAST
  - Basic Local Alignment Search Tool
- Why BLAST
  - Quickly search a sequence database

Alignment in Real Life (25+ years ago)

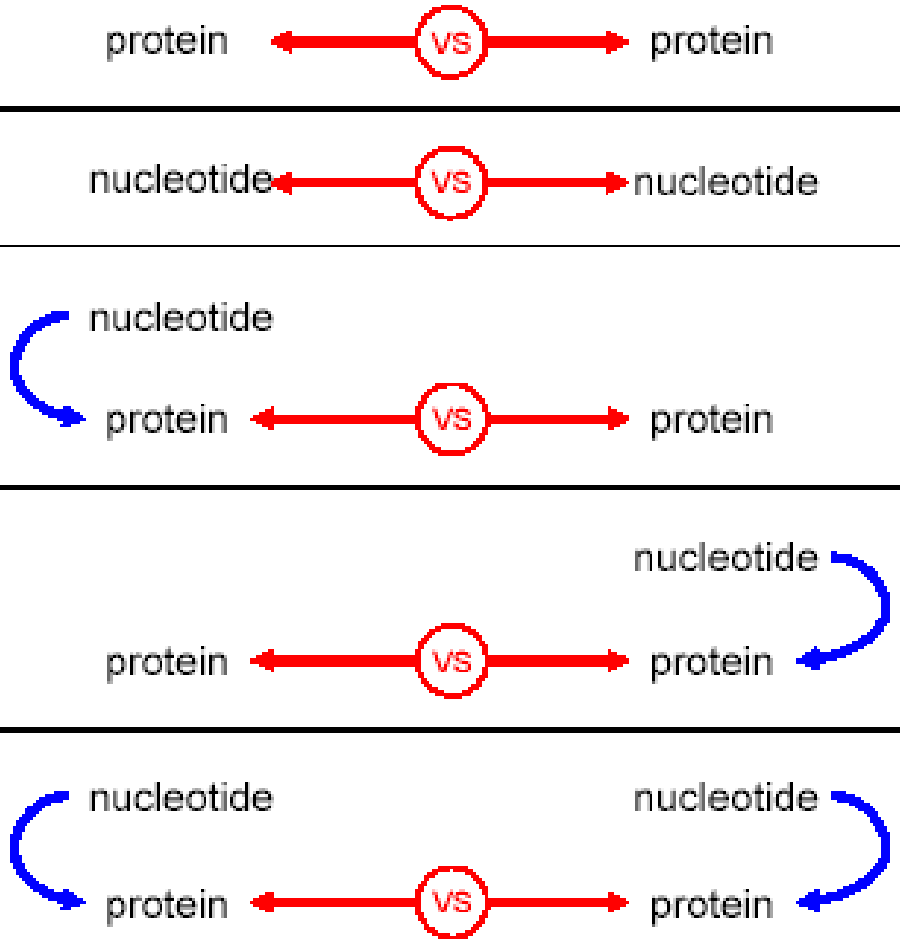
- One of the major uses of alignments is to find sequences in a database
- The current protein database contains about 10^8 residues!
  - Searching a 10^3 base long target sequence requires to evaluate about 10^{11} matrix cells…
  - … which will take about three hours in the rate of 10^7 evaluations per second.
  - Quite annoying when, say, 10^3 sequences are waiting to be searched. About four months will be required for completing the analysis!

Introduction to BLAST

- Different versions of BLAST
  - NCBI-BLAST
  - WU-BLAST (now AB-BLAST)

Different BLAST programs: according to the query and database

Program	Query	Database
blastp	protein	protein
blastn	nucleotide	nucleotide
blastx	nucleotide translated to protein	protein
tblastn	protein	nucleotide translated to protein
tblastx	nucleotide translated to protein	nucleotide translated to protein



Blast output file

```
BLASTP 3.0PE-AB [2009-10-30] [linux26-x64-I32LPF64 2009-11-17T18:52:53]

Copyright (C) 2009 Warren R. Gish.  All rights reserved.
Unlicensed use, reproduction or distribution are prohibited.
Advanced Biocomputing, LLC, licenses this software only for personal use
on a personally owned computer.

Reference:  Gish, W. (1996-2009) http://blast.advbiocomp.com

Query=  RU1A_HUMAN
        (282 letters)

Database:  /home/ccwei/courses/g_and_p/C.elegans/Proteome/ws_215.protein
           24,705 sequences; 10,879,267 total letters.
Searching....10....20....30....40....50....60....70....80....90....100% done

                                                                    Smallest
                                                                      Sum
                                                             High  Probability
Sequences producing High-scoring Segment Pairs:              Score  P(N)      N

K08D10.3        CE07355 WBGene00004386  locus:rnp-3     U1 small nucl...   378  3.2e-53   2
K08D10.4        CE28597 WBGene00004385  locus:rnp-2     U1 small nucl...   332  1.5e-51   2
C50D2.5 CE38492 WBGene00016808  status:Confirmed        UniProt:Q...   113  7.4e-08   1
F46A9.6 CE08260 WBGene00003172  locus:mec-8     mecanosensory ...   111  5.8e-07   2
R09B3.2 CE16307 WBGene00011155  RNA recognition motif. (ak...    91  2.6e-05   1
D2089.4b        CE30509 WBGene00004207  locus:ptb-1     status:Partia...    86  5.4e-05   2
T01D1.2g        CE41586 WBGene00001340  locus:etr-1     status:Confir...    95  6.5e-05   2
T23F6.4 CE18963 WBGene00004315  locus:rbd-1     RNA recognitio...    85  8.1e-05   2
T01D1.2a        CE12942 WBGene00001340  locus:etr-1     RNA-binding p...    95  9.0e-05   2
```

Blast output file

```
>K08D10.3       CE07355 WBGene00004386  locus:rnp-3     U1 small nuclear
ribonucleoprotein
          A     status:Confirmed        UniProt:Q21323  protein_id:AAA98033.1
        Length = 217

 Score = 378 (138.1 bits), Expect = 3.2e-53, Sum P(2) = 3.2e-53
 Identities = 69/116 (59%), Positives = 89/116 (76%)

Query:     5 ETRPNHTIYINNLNEKIKKDELKKSLYAIFSQFGQILDILVSRSLKMRGQAFVIFKEVSS 64
             +  PNHTIY+NNLNEK+KKDELK+SL+ +F+QFG+I+ ++  R  KMRGQA ++FKEVSS
Sbjct:     3 DINPNHTIYVNNLNEKVKKDELKRSLHMVFTQFGEIIQLMSFRKEKMRGQAHIVFKEVSS 62

Query:    65 ATNALRSMQGFPFYDKPMRIQYAKTDSDIIAKMKGTFVXXXXXXXXXXXXSQETPA 120
             A+NALR++QGFPFY KPMRIQYA+ DSD+I++ KGTFV             E PA
Sbjct:    63 ASNALRALQGFPFYGKPMRIQYAREDSDVISRAKGTFVEKRQKSTKIAKKPYEKPA 118

 Score = 179 (68.1 bits), Expect = 3.2e-53, Sum P(2) = 3.2e-53
 Identities = 33/77 (42%), Positives = 49/77 (63%)

Query:   206 PNHILFLTNLPEETNELMLSMLFNQFPGFKEVRLVPGRHDIAFVEFDNEVQAGAARDALQ 265
             PN+ILF +N+PE T    +  +F+QFPG +EVR +P   D AF+E+++E  +  AR AL
Sbjct:   141 PNNILFCSNIPEGTEPEQIQTIFSQFPGLREVRWMPNTKDFAFIEYESEDLSEPARQALD 200

Query:   266 GFKITQNNAMKISFAKK 282
              F+IT    + + FA K
Sbjct:   201 NFRITPTQQITVKFASK 217
```

Heuristic Search

- **Search with clues**
  - **Much faster**
  - **May completely miss the optimal alignment**

- **Two important algorithms**
  - **BLAST**
  - **FASTA**

Basic Intuition 1: Seeds

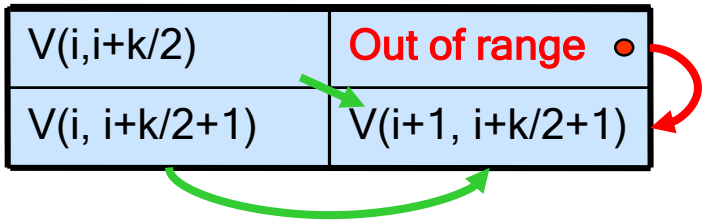
- **Observation: Real-life matches often contain long strings with gap-less matches**

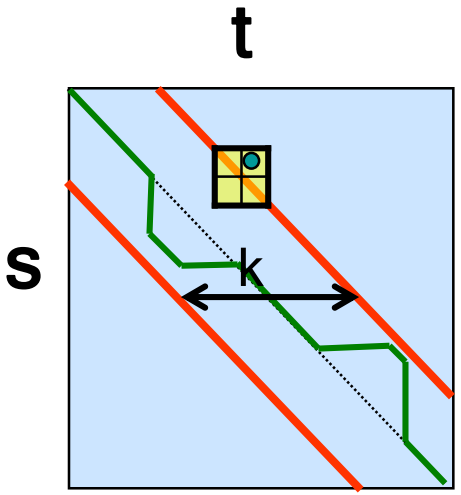
- **Action: Try to find significant gap-less matches and then extend them**

```
>K08D10.3       CE07355 WBGene00004386  locus:rnp-3     U1 small nuclear
ribonucleoprotein
          A     status:Confirmed        UniProt:Q21323  protein_id:AAA98033.1
        Length = 217

 Score = 378 (138.1 bits), Expect = 3.2e-53, Sum P(2) = 3.2e-53
 Identities = 69/116 (59%), Positives = 89/116 (76%)

Query:     5 ETRPNHTIYINNLNEKIKKDELKKSLYAIFSQFGQILDILVSRSLKMRGQAFVIFKEVSS 64
             +  PNHTIY+NNLNEK+KKDELK+SL+ +F+QFG+I+ ++  R  KMRGQA ++FKEVSS
Sbjct:     3 DINPNHTIYVNNLNEKVKKDELKRSLHMVFTQFGEIIQLMSFRKEKMRGQAHIVFKEVSS 62

Query:    65 ATNALRSMQGFPFYDKPMRIQYAKTDSDIIAKMKGTFVXXXXXXXXXXXXSQETPA 120
             A+NALR++QGFPFY KPMRIQYA+ DSD+I++ KGTFV            E PA
Sbjct:    63 ASNALRALQGFPFYGKPMRIQYAREDSDVISRAKGTFVEKRQKSTKIAKKPYEKPA 118
```

Basic Intuition 2: Banded DP

- **Observation: If the optimal alignment of `s` and `t` has few gaps, then path of the alignment will be close to diagonal**



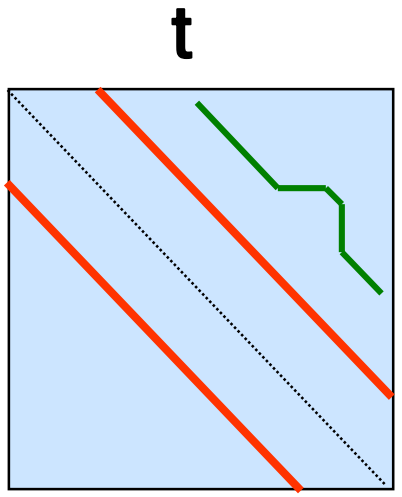


- **Action: Search in a diagonal band of the matrix.**
  - **If the diagonal band consists of `k` diagonals (width `k`), then dynamic programming takes O(`kn`).**
  - **Much faster than $O(n^2)$ of standard DP.**

Banded DP for Local Alignment

- **Problem: The banded diagonal needs not be the main diagonal when looking for a good local alignment**
  - **Also the case when the lengths of `s` and `t` are different**

- **Solution: Heuristically find potential diagonals and evaluate them using Banded DP**



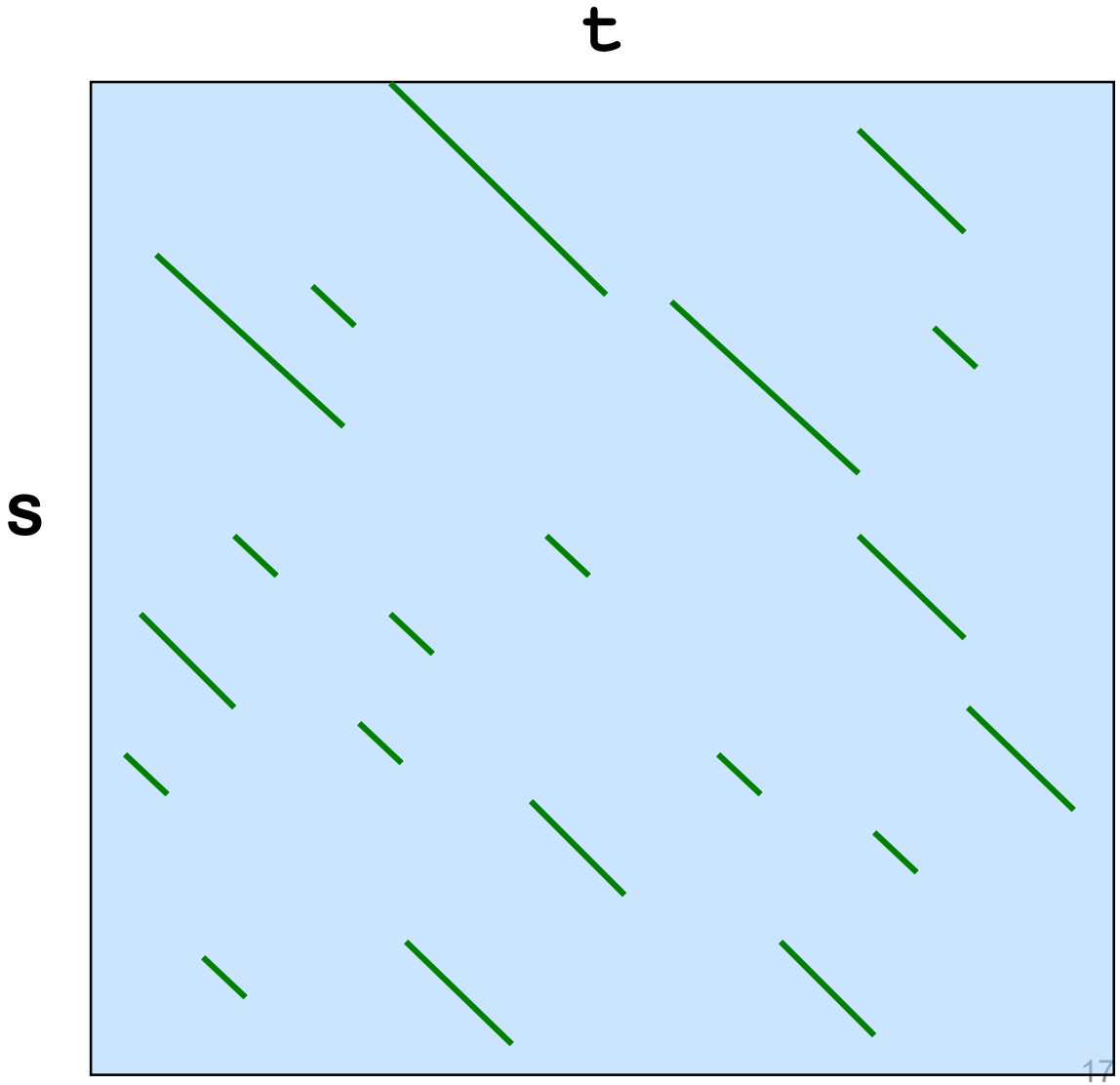
FASTA

- **Publication**
  - **Pearson and Lipman, 1988**
- **Input**
  - **Two sequences `s` and `t`**
  - **Parameter `ktup` – defines the length of seeds.**
    - Typically **`ktup`**=1-2 for proteins and **`ktup`**=4-6 for DNA/RNA
- **Output**
  - **The best local alignment between `s` and `t`**

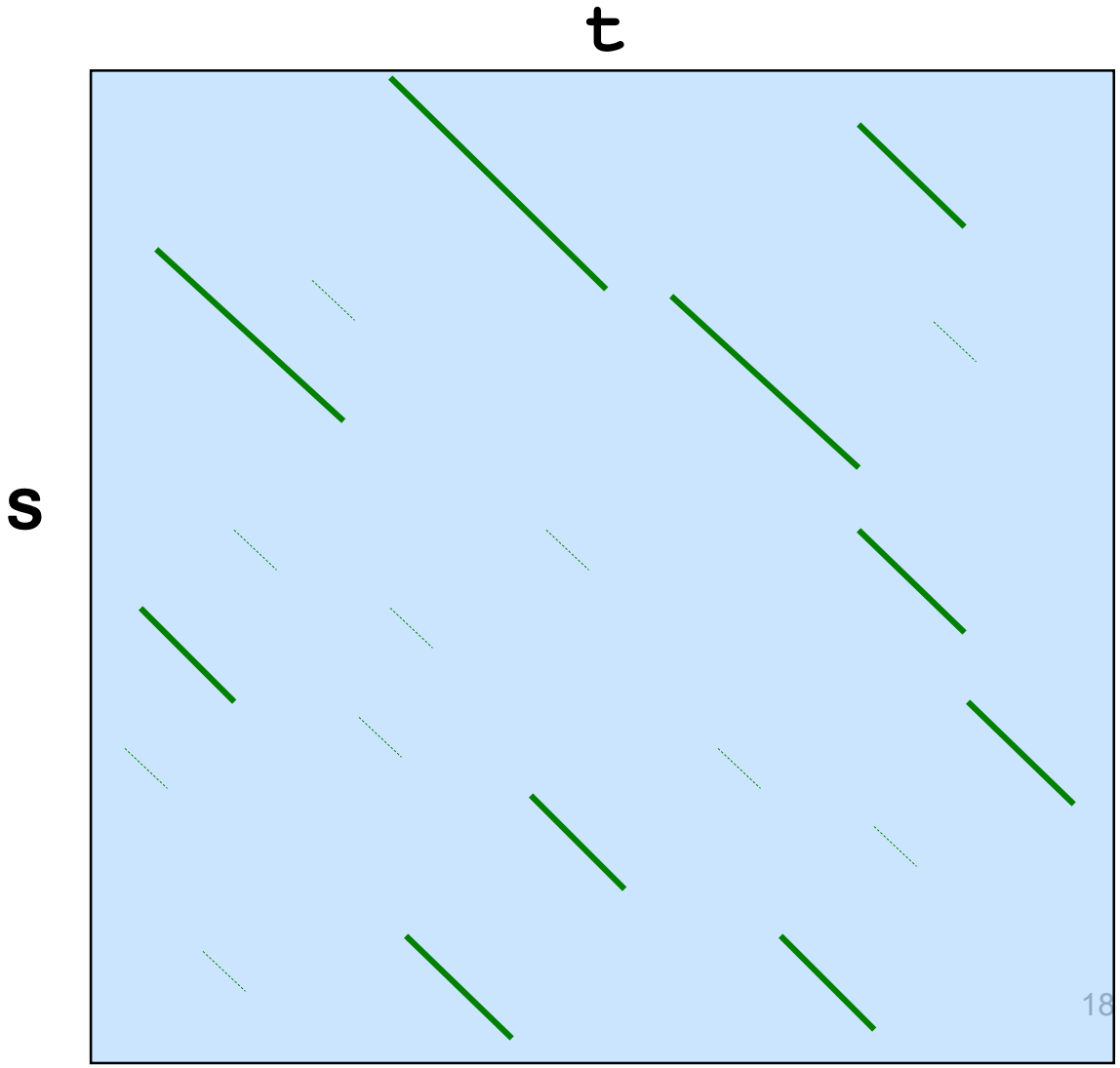
FASTA – Algorithm Outline

1. Find regions in `s` and `t` containing high density of seeds
2. Re-score the 10 regions with the highest scores using PAM matrix
3. Eliminate segments that are unlikely to be part of alignments
4. Optimize the best alignment using the banded DP algorithm

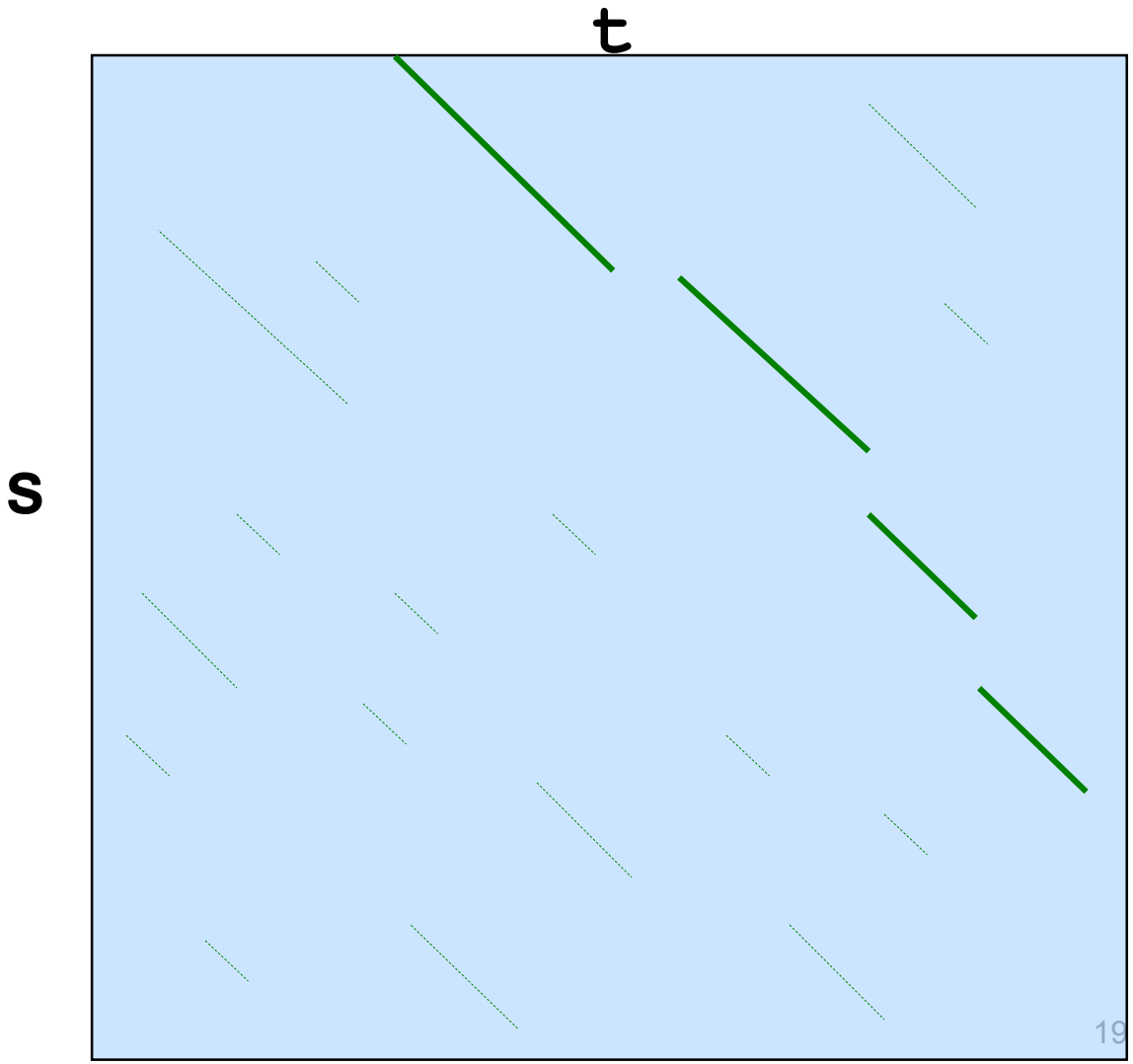
Step 1: Finding Seeds



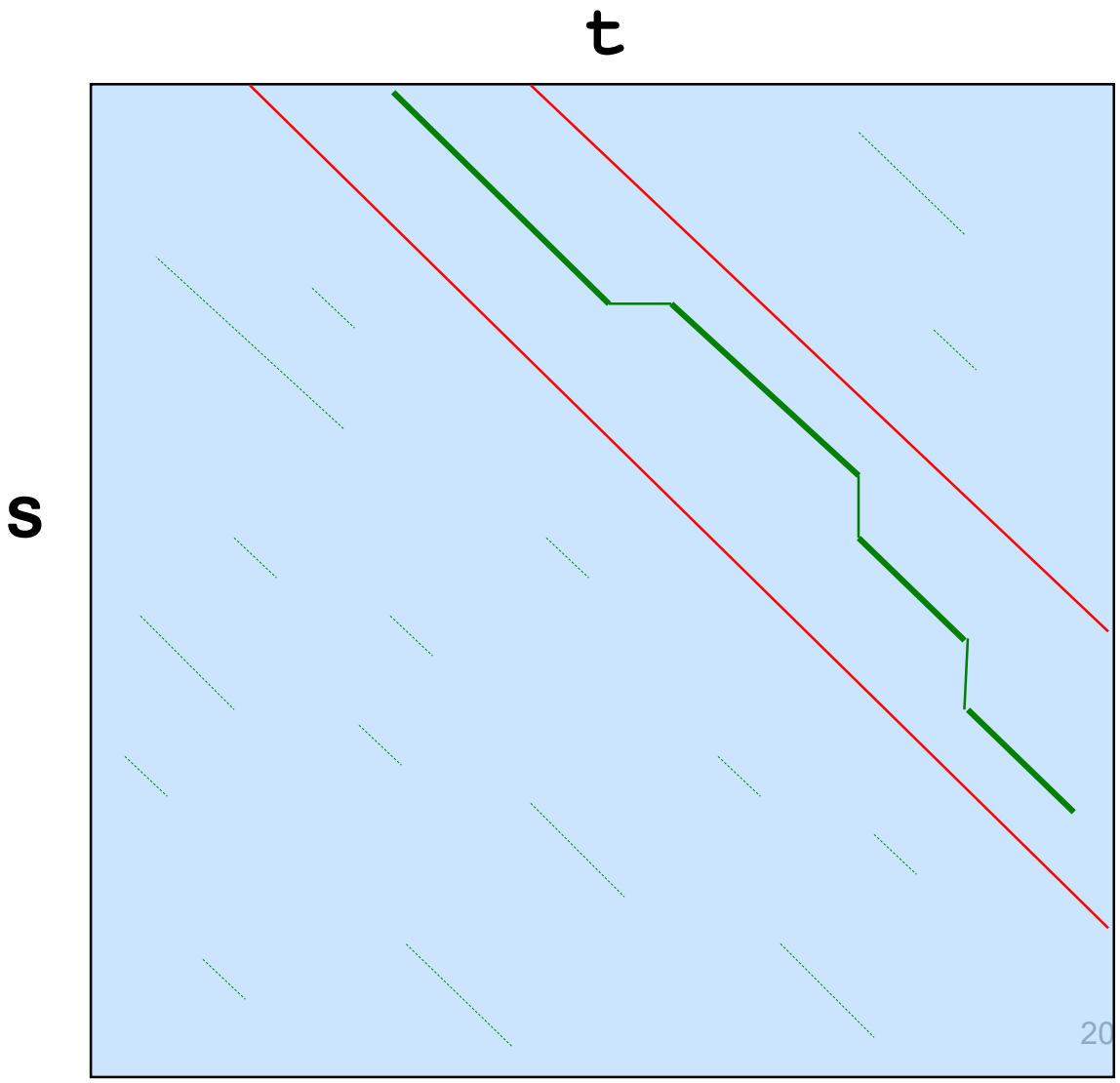
Step 2: Re-scoring Segments, Keeping Top 10



Step 3: Eliminating Unlikely Segments



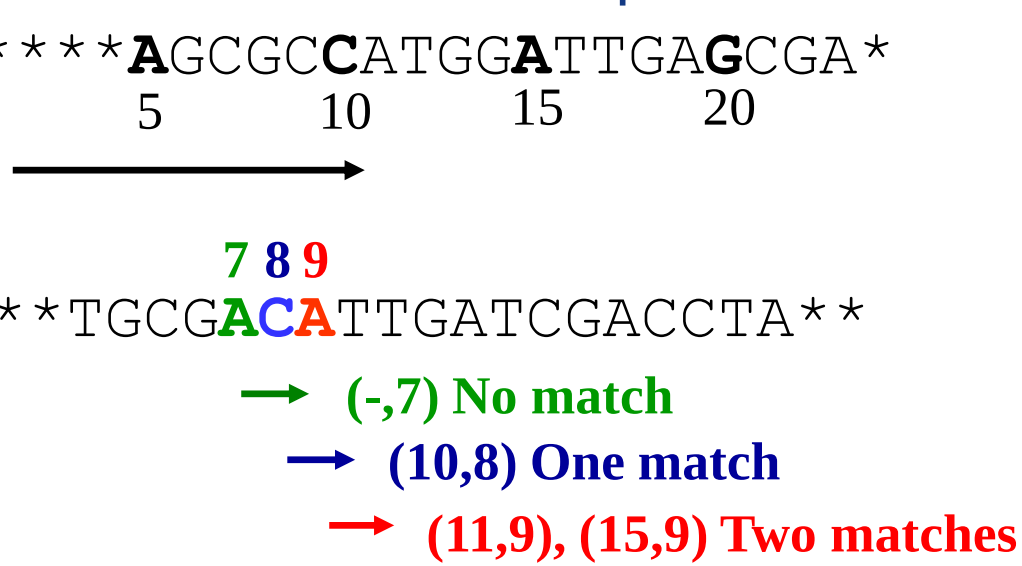
Step 4: Finding the Best Alignment



Finding Seeds Efficiently

- Prepare an index table of the database sequence s such that for any sequence of length ktup, one gets the list of its positions in s.
- March on the query sequence t while using the index table to list all matches with the database sequence s.

Index Table (ktup=2)	
AA	-
AC	-
AG	5, 19
AT	11, 15
CA	10
CC	9
CG	7 , 21
…	
TT	16



Connecting Seeds on the Same Diagonal

- **The maximal size of the index table is $|\sum|^{ktup}$, where $\sum$ is the alphabet size (4 or 20).**
  - **For small ktup, the entire table is stored**
  - **For large ktup values**
    - only entries for tuples actually found in the database
    - In this case, hashing is needed
- **Typical values of ktup**
  - **1-2 for Proteins**
  - **4-6 for DNA**
- **The index table is prepared for each database sequence ahead of users' matching requests, at compilation time.**
  - **Matching time is O(|t|·max{row_length})**

Identifying Potential Diagonals

- **Input: Sets of pairs**
  - **E.g, (6,4),(10,8),(14,12),(15,10),(20,4) …**
- **Task**
  - **Locate sets of pairs that are on the same diagonal.**
- **Method**
  - **Sort according to the difference *i-j.***
  - **E.g, 6-4=2, 10-8=2, 14-12=2, 15-10=5, 20-4=16 …**

FASTA Parameters

- *ktup* = 2 for proteins, 6 for DNA
- *init1* Score after rescanning with PAM250 (or other)
- *initn* Score after joining regions
- *opt* Score after Banded DP

Limits

- **Local similarity might be missed because only 10 regions saved at *init1* stage.**
- **Non-identical conserved stretches may be overlooked**

Basic Local Alignment Search Tool (BLAST)

- **Publications:**
  - Ungapped BLAST – Altschul et al., 1990
  - Gapped BLAST, PSI-BLAST - Altschul et al., 1997
- **Input:**
  - Query (target) sequence – either DNA, RNA or Protein
  - Scoring Scheme – gap penalties, substitution matrix for proteins, identity/mismatch scores for DNA/RNA
  - Word length `w` – typical is `w`=3 for proteins and `w`=11 for DNA/RNA
- **Output:**
  - Statistically significant matches

PART II inside into BLAST

Mathematic model of sequence alignment

Alphabet of biological sequence

- Nucleic acid sequence

  {A,T,C,G}

- Amino acid sequence

  {A,S,G,L,K,V,T,P,E,D,N,I,Q,R,F,Y,C,H,M,W}

Operation of sequence alignment

- Match (A,A)
- Replace (A,T)
- Delete (A, -)
- Insert (- , A)

Mathematic model of sequence alignment

How to define similarity between two sequences?

Distance

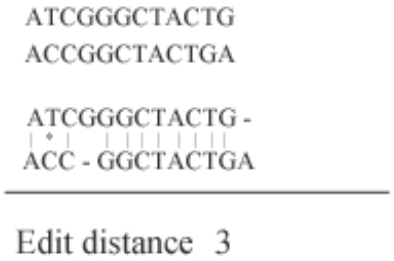
- **Hamming distance**

Mismatch number of two sequences with same length

- **Edit distance**

Operation number for one sequence transforming to another

s =	AAT	AGCAA	AGCACACA
t =	TAA	ACATA	ACACACTA
Hamming Distance(s,t)=	2	3	6



Mathematic model of sequence alignment

How to quantify the distance

Scoring

Simple scoring function

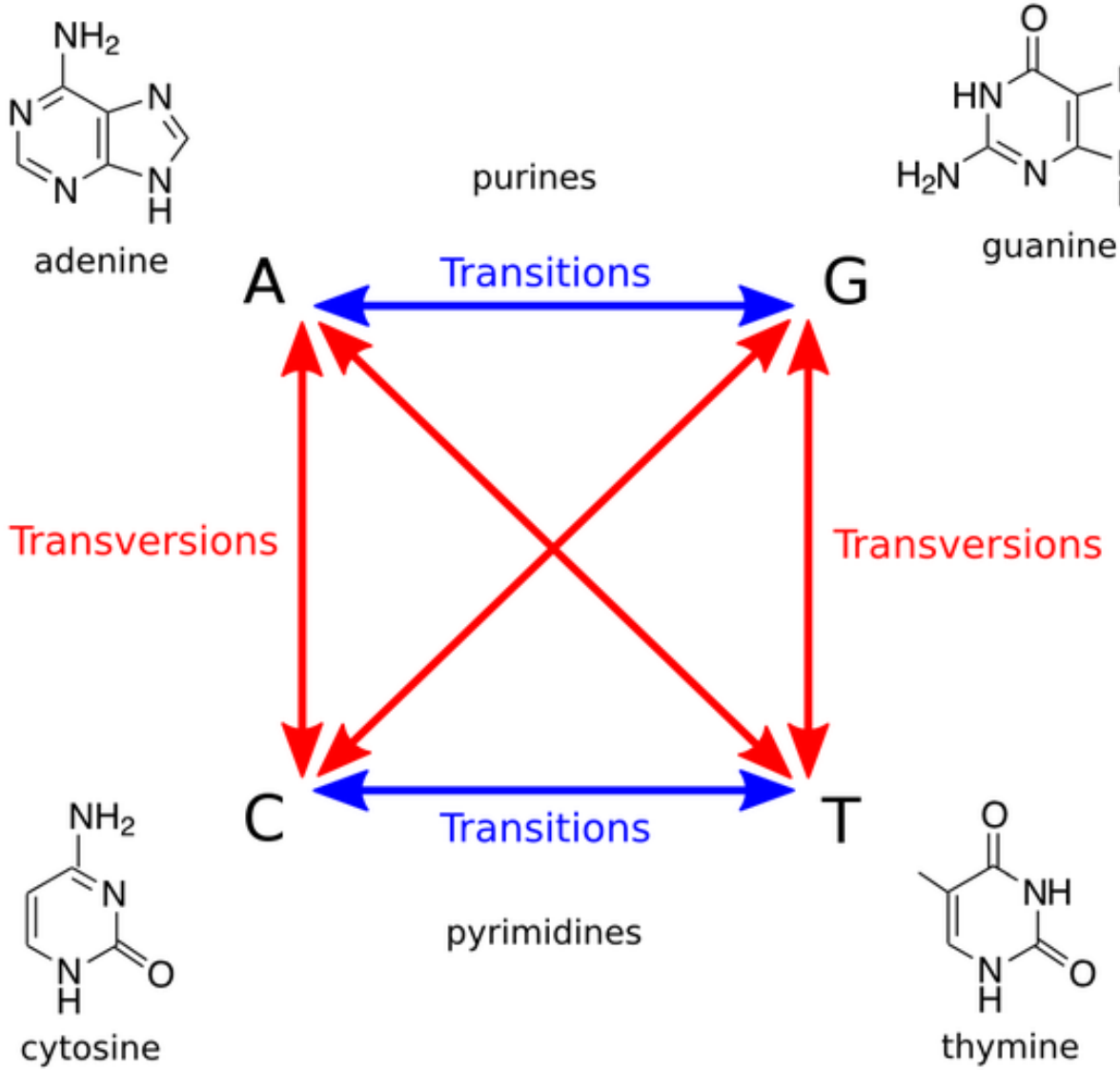
$$\begin{cases} \text{Match}(A, A) = 1 \\ \text{Substitution}(A, T) = 0 \\ \text{Delete}(A, -) = \text{Insert}(-, A) = -1 \end{cases}$$

Matrix for scoring

Matrix for nucleic acid sequence alignment

Matrix for amino acid sequence alignment

Mathematic model of sequence alignment



Transition-transversion matrix

	A	T	C	G
A	1	-5	-5	-1
T	-5	1	-1	-5
C	-5	-1	1	-5
G	-1	-5	-5	1

Mathematic model of sequence alignment

Matrix for amino acid sequence alignment

(1) identity matrix

(2) Point accepted mutation matrix (PAM)

(3) BLOSUM matrix

Mathematic model of sequence alignment

PAM70

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-4	-2	-1	-4	-2	-1	0	-4	-2	-4	-4	-3	-6	0	1	1	-9	-5	-1	-1	-1	-2	-11
R	-4	8	-3	-6	-5	0	-5	-6	0	-3	-6	2	-2	-7	-2	-1	-4	0	-7	-5	-4	-2	-3	-11
N	-2	-3	6	3	-7	-1	0	-1	1	-3	-5	0	-5	-6	-3	1	0	-6	-3	-5	5	-1	-2	-11
D	-1	-6	3	6	-9	0	3	-1	-1	-5	-8	-2	-7	-10	-4	-1	-2	-10	-7	-5	5	2	-3	-11
C	-4	-5	-7	-9	9	-9	-9	-6	-5	-4	-10	-9	-9	-8	-5	-1	-5	-11	-2	-4	-8	-9	-6	-11
Q	-2	0	-1	0	-9	7	2	-4	2	-5	-3	-1	-2	-9	-1	-3	-3	-8	-8	-4	-1	5	-2	-11
E	-1	-5	0	3	-9	2	6	-2	-2	-4	-6	-2	-4	-9	-3	-2	-3	-11	-6	-4	2	5	-3	-11
G	0	-6	-1	-1	-6	-4	-2	6	-6	-6	-7	-5	-6	-7	-3	0	-3	-10	-9	-3	-1	-3	-3	-11
H	-4	0	1	-1	-5	2	-2	-6	8	-6	-4	-3	-6	-4	-2	-3	-4	-5	-1	-4	0	1	-3	-11
I	-2	-3	-3	-5	-4	-5	-4	-6	-6	7	1	-4	1	0	-5	-4	-1	-9	-4	3	-4	-4	-3	-11
L	-4	-6	-5	-8	-10	-3	-6	-7	-4	1	6	-5	2	-1	-5	-6	-4	-4	-4	0	-6	-4	-4	-11
K	-4	2	0	-2	-9	-1	-2	-5	-3	-4	-5	6	0	-9	-4	-2	-1	-7	-7	-6	-1	-2	-3	-11
M	-3	-2	-5	-7	-9	-2	-4	-6	-6	1	2	0	10	-2	-5	-3	-2	-8	-7	0	-6	-3	-3	-11
F	-6	-7	-6	-10	-8	-9	-9	-7	-4	0	-1	-9	-2	8	-7	-4	-6	-2	4	-5	-7	-9	-5	-11
P	0	-2	-3	-4	-5	-1	-3	-3	-2	-5	-5	-4	-5	-7	7	0	-2	-9	-9	-3	-4	-2	-3	-11
S	1	-1	1	-1	-1	-3	-2	0	-3	-4	-6	-2	-3	-4	0	5	2	-3	-5	-3	0	-2	-1	-11
T	1	-4	0	-2	-5	-3	-3	-3	-4	-1	-4	-1	-2	-6	-2	2	6	-8	-4	-1	-1	-3	-2	-11
W	-9	0	-6	-10	-11	-8	-11	-10	-5	-9	-4	-7	-8	-2	-9	-3	-8	13	-3	-10	-7	-10	-7	-11
Y	-5	-7	-3	-7	-2	-8	-6	-9	-1	-4	-4	-7	-7	4	-9	-5	-4	-3	9	-5	-4	-7	-5	-11
V	-1	-5	-5	-5	-4	-4	-4	-3	-4	3	0	-6	0	-5	-3	-3	-1	-10	-5	6	-5	-4	-2	-11
B	-1	-4	5	5	-8	-1	2	-1	0	-4	-6	-1	-6	-7	-4	0	-1	-7	-4	-5	5	1	-2	-11
Z	-1	-2	-1	2	-9	5	5	-3	1	-4	-4	-2	-3	-9	-2	-2	-3	-10	-7	-4	1	5	-3	-11
X	-2	-3	-2	-3	-6	-2	-3	-3	-3	-3	-4	-3	-3	-5	-3	-1	-2	-7	-5	-2	-2	-3	-3	-11
*	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	-11	1

How to create PAMs

PAM1=substitution matrix for aas mutation rate of 1%

PAM2=PAM1*PAM1

…

PAMN=PAM1^n

How to create BLOSUM

Clustering proteins with similarity above a certain threshold, then the substitution rates were counted from the multiple alignment

BLOck Substitution Matrix: BLOSUM

Mathematic model of sequence alignment

BLOSUM 62

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

Algorithm of BLAST

- Find high-scoring segment pairs (HSP) contained in a statistically significant alignment.
- Using a heuristic approach that approximates the Smith-Waterman algorithm
- Not optimal, but over 50 times faster than Smith-Waterman

BLAST – Algorithm Outline

1. **Listing seeds**
   - words of length `W` that score at least `T` when aligned with the query sequence `s`
2. **Extracting seeds**
   - search the database `DB` for seeds
3. **Finding High Scoring Pairs (HSPs)**
   - Extend the seeds in both directions. Keep best scoring HSPs
4. **Combine HSPs**
   - banded DP algorithm

Step 1: Listing High Scoring Words of Length w

- **Word length w=3 and score≥T**

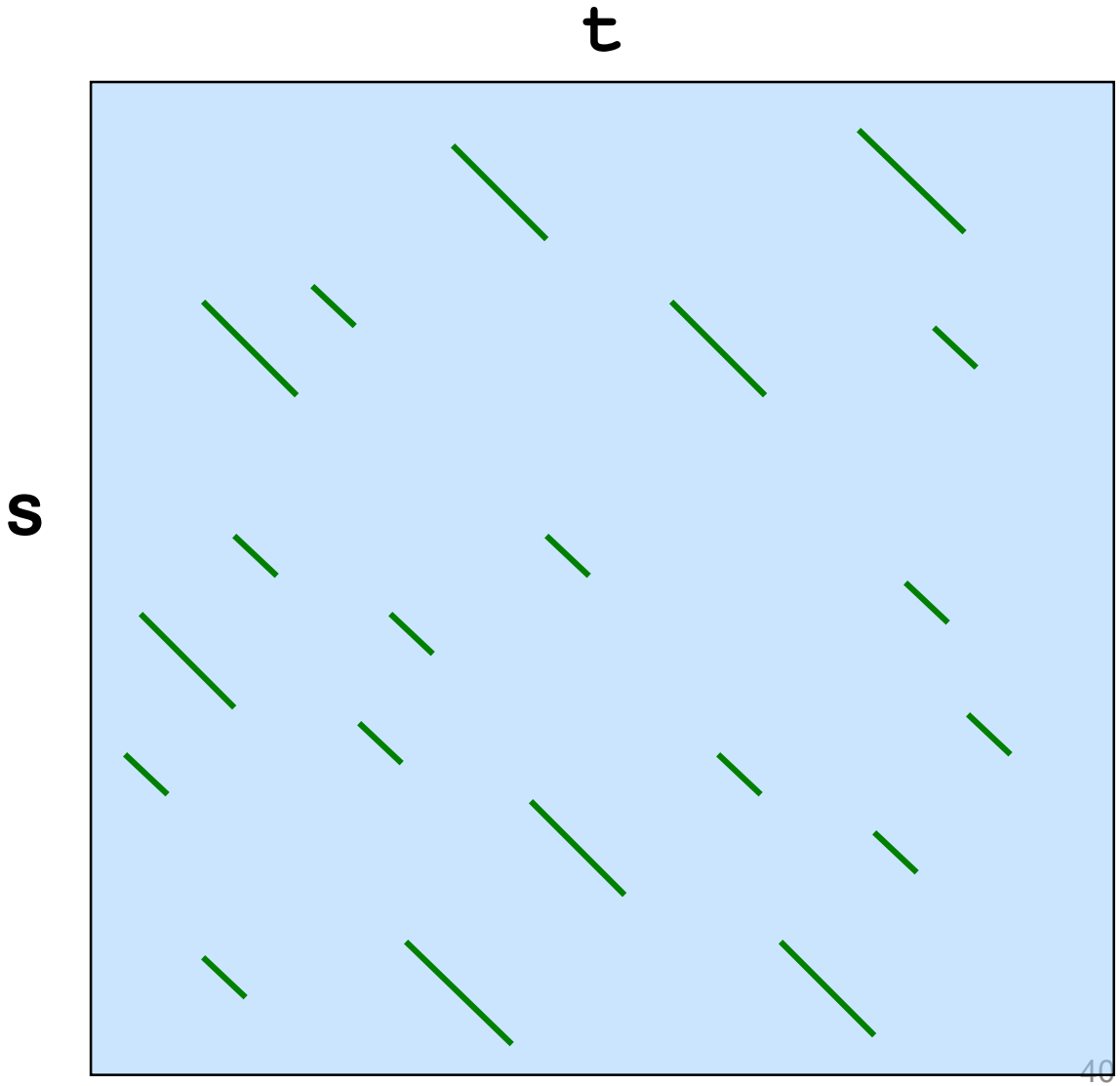
...GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDK...

High scoring words

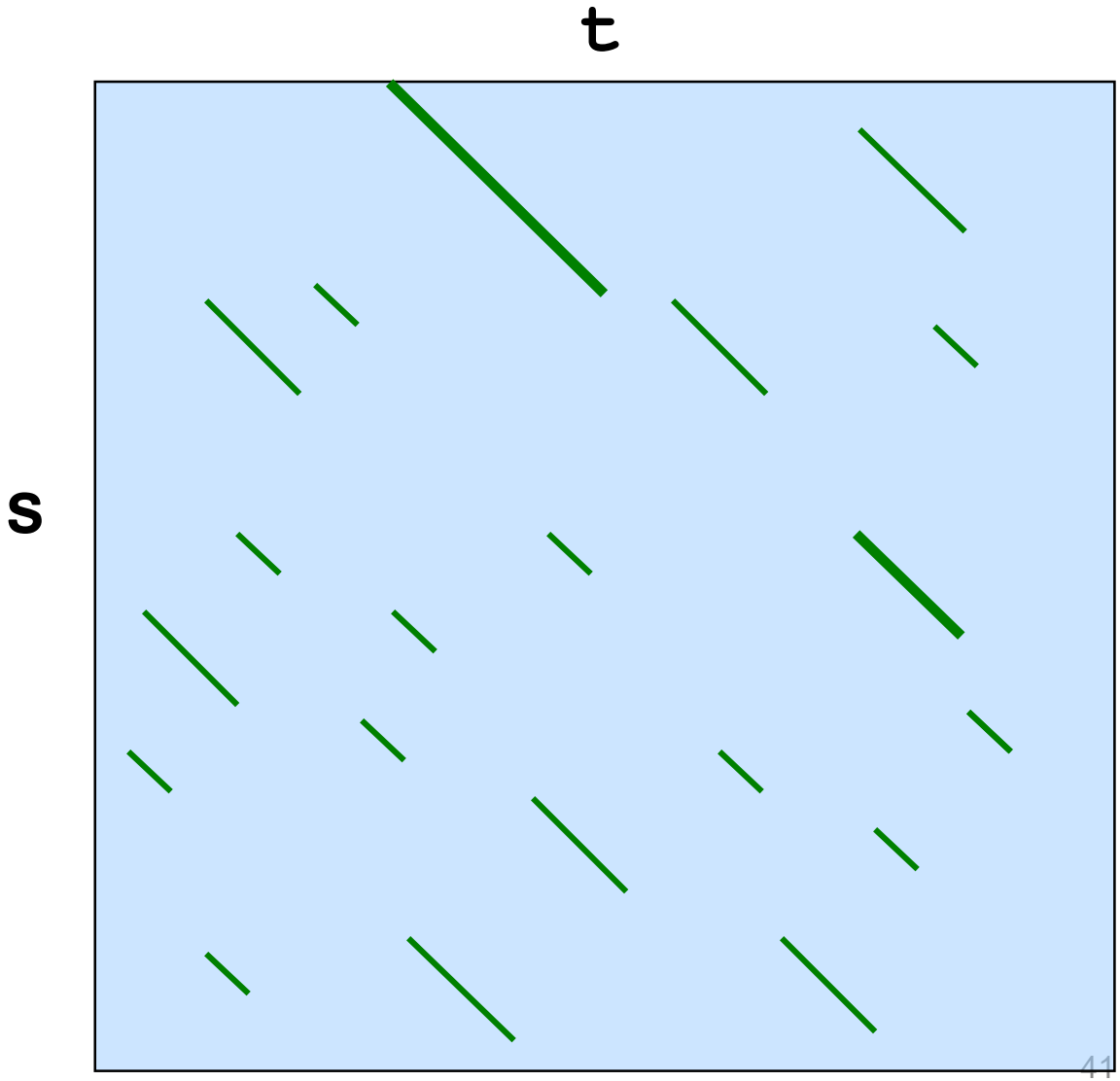
PQG	18
PEG	15
PRG	14
PKG	14
PNG	13
PDG	13
PHG	13
PMG	13
PSG	13

Score threshold
T=13

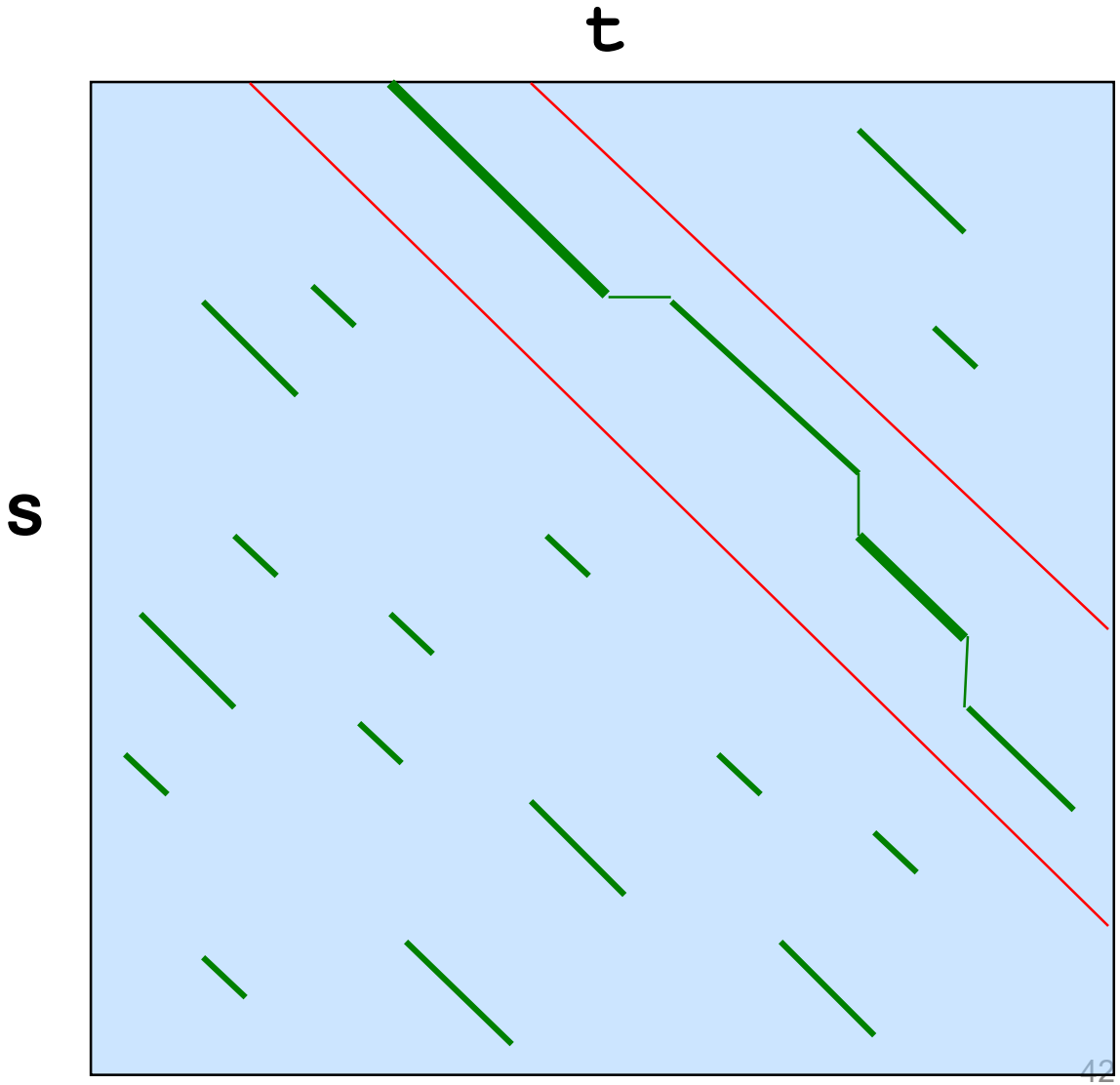
Step 2: Extracting Seeds



Step 3: Finding HSPs



Step 4: Combining HSPs



BLAST – Notes

- Seed parameters（W, T）
  - Higher W or T → lower sensitivity, runs faster
- Extracting seeds
  - Use hash tables to make the process faster
- Finding HSPs
  - Only seeds located on the same diagonal with some other seeds located at a distance smaller than a threshold will be extended
- Gapped alignment
  - Will be triggered only for HSPs whose scores are higher than the threshold

Karlin-Altschul statistics

If we search two sequence X and Y with a scoring matrix $s_{i,j}$ for maximal-scoring segment pair, and if the following conditions hold:

1. Letters of the two sequences are both i.i.d. with distributions P_x and P_y (can be the same);
2. Both sequences are long enough;
3. The expected pairwise score $\sum_{i,j} p_x(i)p_y(j)s_{i,j}$ is negative;
4. A positive score is possible, i.e. $P_x(i)P_y(j) > 0$ for some i and j.

Karlin-Altschul statistics tell us:

Karlin-Altschul statistics

- The maximal segment score has the approximating distribution:

$$Prob(S > x) \approx= 1 - \exp(-K * \exp^{-\lambda * x})$$

where K and λ are constants that can be calculated according to

Karlin, S, and SF Altschul (1990), "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes", PNAS 87:2264-68

Karlin-Altschul statistics

- The expected number of occurrences of an MSP with score S or greater by chance is:

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

Karlin-Altschul statistics

- The scores in the scoring matrix are implicitly log-odds scores of the form:

$$S_{ij} = \log(Q_{ij} / (P_X(i) P_Y(j))) / \lambda$$

where Q_{ij} is the limiting target distribution of the letter pairs (i,j) in the MSP and λ is the unique positive-valued solution to the equation

$$\sum_{i,j} P_X(i) P_Y(j) e^{\lambda S_{ij}} = 1$$

Karlin-Altschul statistics

- Another way to express the scores in the scoring matrix:

$$S_{ij} = \log_b(Q_{ij}/(P_X(i)P_Y(j)))$$

where logarithms to some base b are used instead of Natural logarithms. Then λ is related to the base of the logarithms as follows:

$$\lambda \log_e b = 1$$

- The expected length of the MSP is

$$E(L) = \log(KMN)/H$$

where H is the relative entropy of the target and background frequencies:

$$H = \sum_{i,j}(Q_{ij}\log(Q_{ij}/(P_X(i)P_Y(j))))$$

Karlin-Altschul statistics

- The expect score E of a database match is the number of times that an unrelated database sequence would obtain a score of S or higher by chance. (The relationship of P-value and E-value)

$$P \approx 1 - e^{-E}$$

- Normalized score for different database search

$$S' = \lambda S - \log K$$

then,

$$E = MNe^{-S'}$$

Notes about the scores in Blast

- **What does a big score mean?**

Acknowledgement

- Some of the slides are from Dr. Guangyong Zheng, CAS