In the Name of God, the Merciful, the Compassionate

Introduction to Bioinformatics
05: Database Similarity Searching

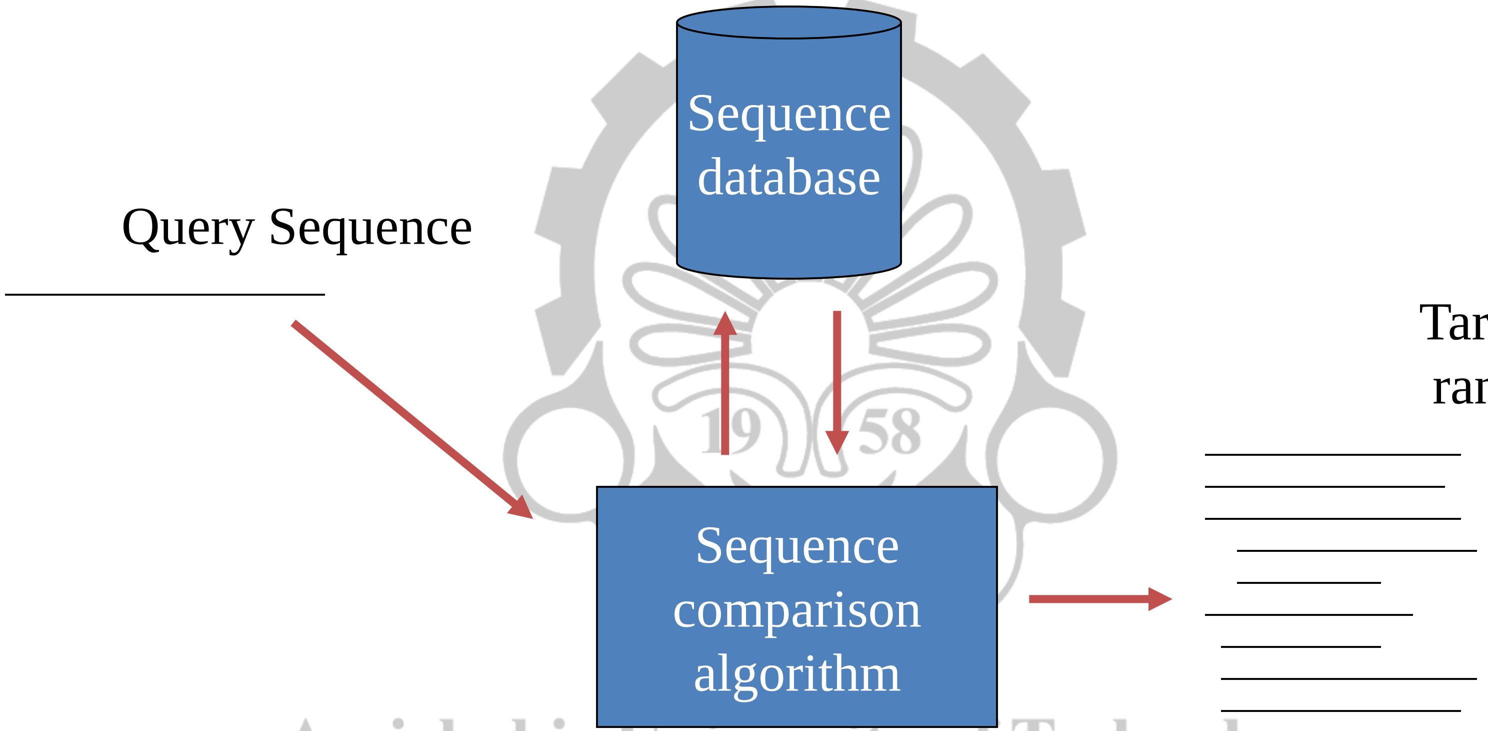
Instructor: Hossein Zeinali

Amirkabir University of Technology

Chapter Agenda

- Unique Requirements of Database Searching
- Heuristic Database Searching
- Basic Local Alignment Search Tool (BLAST)
- FASTA
- Comparison of FASTA and BLAST
- Database Searching with Smith-Waterman Method

Database searching



Sequence database

Query Sequence

Sequence comparison algorithm

Target sequences ranked by score

Why database search is needed?

- Given a newly discovered gene,
  - Does it occur in other species?
  - Is its function known in another species?

- Given a newly sequenced genome, which regions align with genomes of other organisms?
  - Identification of potential genes
  - Identification of other functional parts of chromosomes

- Find members of a multigene family

Why do we Need Fast Search Algorithms?

- Your query is 200 amino acids (aa) long (*N*)
- You are searching a non-redundant database, which currently contains $>10^6$ proteins (*K*)
- If proteins in database have avg. length 200 aa (*M*), then:
  - Must fill in $200 \times 200 \times 10^6 =$ **4×10^{10} DP entries!!**
- 4×10^{10} operations just to *fill in* the DP matrix!

- DP for pairwise alignment is **O(NM)**
- Searching in a database is **O(*NMK*)**
  - **Need *faster* algorithms for searching in large databases!**
- ***Speed*** is the time it takes to get results from database searches.

Sensitivity and Specificity

- *Sensitivity (Recall)*: the ability to find as many correct hits as possible and measures the proportion of actual positives that are correctly identified as such.
- *Specificity (Selectivity)*: the ability to exclude incorrect hits and measures the proportion of actual negatives that are correctly identified as such.
- Example in disease: **positive** means having the disease and **negative** means not having the disease.
  - True positive (TP): Sick people correctly identified as sick
  - False positive (FP): Healthy people incorrectly identified as sick
  - True negative (TN): Healthy people correctly identified as healthy
  - False negative (FN): Sick people incorrectly identified as healthy

TP eqv Hit
FP eqv False alarm
TN eqv Correct rejection
FN eqv Miss

Sensitivity and Specificity (Cont.)

- *Sensitivity (Recall)*: the ability to find as many correct hits as possible.

$$Sensitivity = \frac{TP}{TP + FN}$$

- *Specificity (Selectivity)*: the ability to exclude incorrect hits.

$$Specificity = \frac{TN}{TN + FP}$$

- The ideal case is having the greatest sensitivity, selectivity, and speed in database searches.

Sensitivity and S

t.)

- *Sensitivity (Recall)*:

any correct hits as

possible.

S

$\overline{}$

N

- *Specificity (Selectiv*

ude incorrect hits.

S

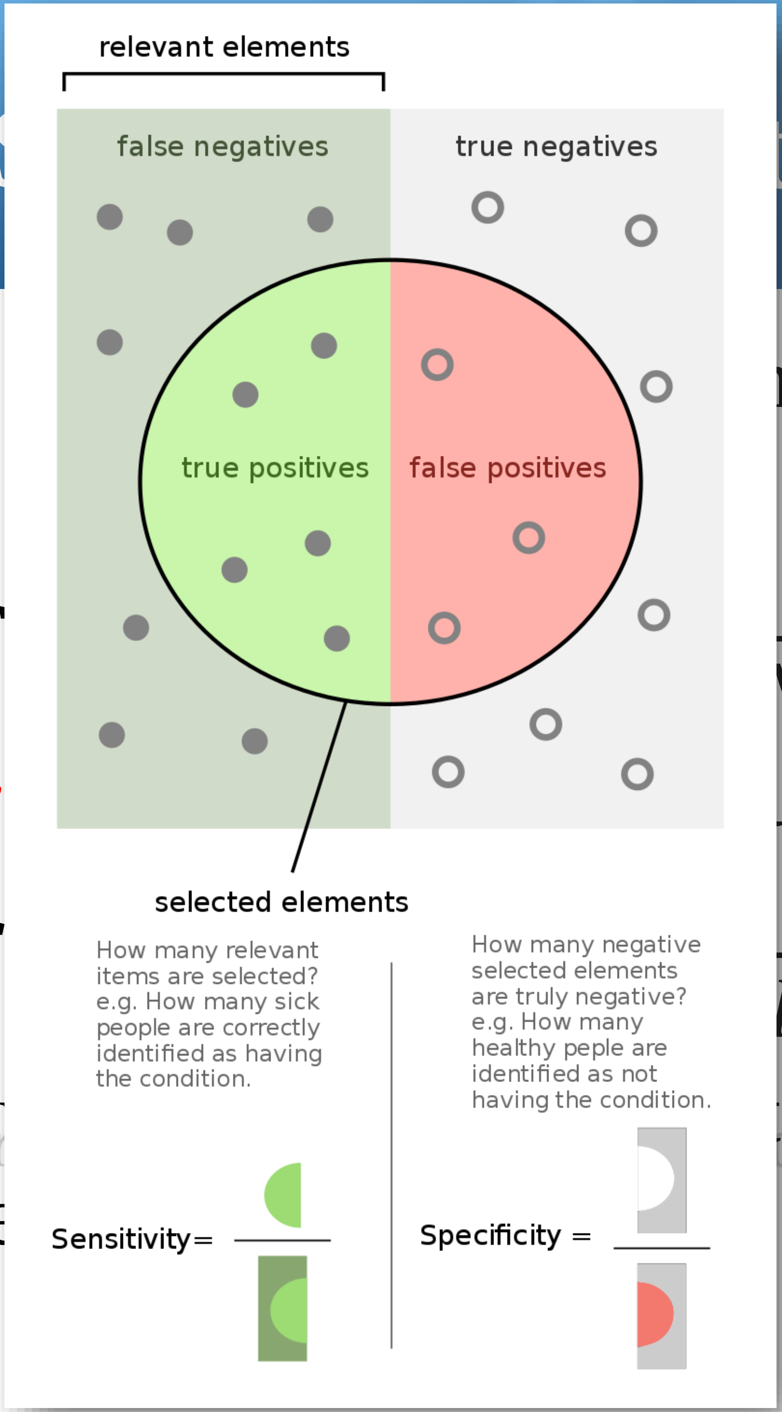
$\overline{}$

P

- The ideal case is ha

ivity, selectivity, and

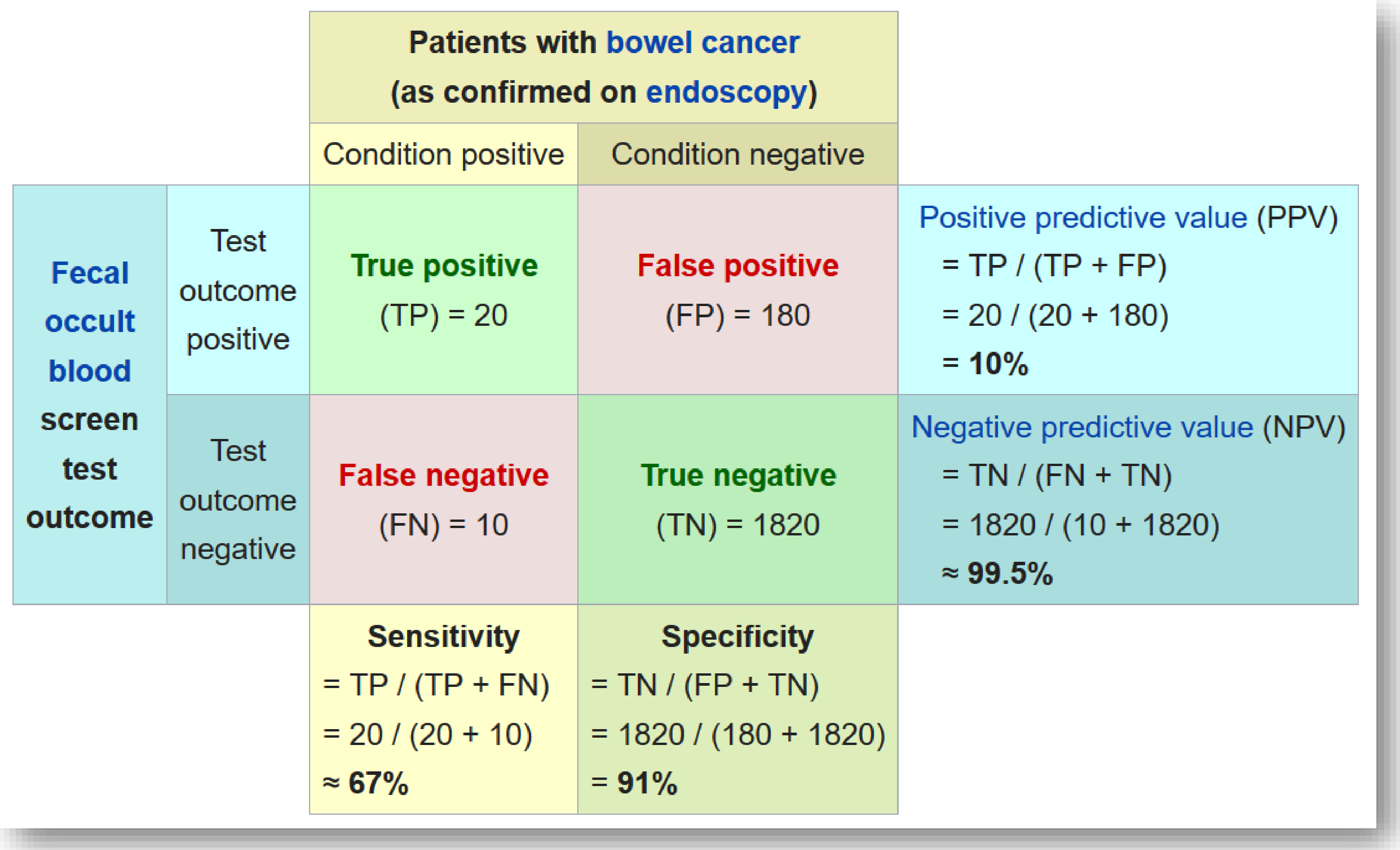
speed in database se



How many relevant items are selected? e.g. How many sick people are correctly identified as having the condition.

How many negative selected elements are truly negative? e.g. How many healthy peple are identified as not having the condition.

Example



		Patients with bowel cancer (as confirmed on endoscopy)		
		Condition positive	Condition negative	
Fecal occult blood screen test outcome	Test outcome positive	**True positive** (TP) = 20	**False positive** (FP) = 180	Positive predictive value (PPV) = TP / (TP + FP) = 20 / (20 + 180) = **10%**
	Test outcome negative	**False negative** (FN) = 10	**True negative** (TN) = 1820	Negative predictive value (NPV) = TN / (FN + TN) = 1820 / (10 + 1820) ≈ **99.5%**
		Sensitivity = TP / (TP + FN) = 20 / (20 + 10) ≈ **67%**	**Specificity** = TN / (FP + TN) = 1820 / (180 + 1820) = **91%**	

Exhaustive vs Heuristic Methods

- ***Exhaustive***
  - Tests every possible solution
  - Guaranteed to give best answer (identifies optimal solution)
  - Can be very time/space intensive!
    - e.g., ***Dynamic Programming*** (as in Smith-Waterman algorithm)
  - Example: querying a database of 300,000 sequences using a query sequence of 100 residues took 2–3 hours to complete.
- ***Heuristic***
  - Does NOT test every possibility
  - No guarantee that answer is best (but, often can identify optimal solution, 50–100 times faster with a moderate expense of sensitivity and specificity)
  - Sacrifices accuracy (potentially) for speed
  - Uses "rules of thumb" or "shortcuts“
    - e.g., ***BLAST*** & ***FASTA*** which use a heuristic *word method*

FASTA vs BLAST

- Both FASTA, BLAST are based on heuristics
- **Tradeoff: Sensitivity *vs* Speed**
- DP is slower, but more sensitive

- **FASTA**
  - User defines value for ***k* = word length**
  - Slower, but more sensitive than BLAST at lower values of *k*, (preferred for searches involving a very short query sequence)
- **BLAST family**
  - Family of different algorithms *optimized* for particular types of queries, such as searching for distantly related sequence matches
  - *BLAST was developed to provide a faster alternative to FASTA without sacrificing much accuracy*

Basic Local Alignment Search Tool (BLAST)

Steps in BLAST

1. Create list of very possible "word" (e.g., 3-11 residues) from query sequence (Seeding)
2. Search database to identify sequences that contain matching words (Searching)
3. The *matching* of the *words* is scored by a given *substitution matrix*.
4. Extend match (seed) in both directions using pairwise alignment, while calculating alignment score at each step (Extension)
5. Continue extension until score drops below a *threshold* (due to mismatches).

High Scoring Segment Pair (HSP) - the resulting contiguous aligned segment pair without gaps.

What are the Results of a BLAST Search?

- Original version of BLAST?
  - – List of **HSPs** called **Maximum Scoring Pairs**

- More recent, improved version of BLAST?
  - – Allows gaps: **Gapped Alignment**
  - – **How**? Allows score to drop below threshold, (but only temporarily)

Why is Gapped Alignment Harder?

- Without gaps, there are N+M-1 possible alignments between sequences of length N and M
- Once we start allowing gaps, there are many more possible arrangements to consider:

```
abcbcd     abcbcd     abcbcd
|||  |     |  |||     ||  ||
abc--d     a--bcd     ab--cd
```

- Becomes a very large number when we also allow mismatches, because we need to look at every possible pairing between elements:

Roughly N^M possible alignments!

e.g.: for N=M=100, there are $100^{100}=10^{200}$ possible alignments & 100 aa is a small protein!

BLAST - a few details

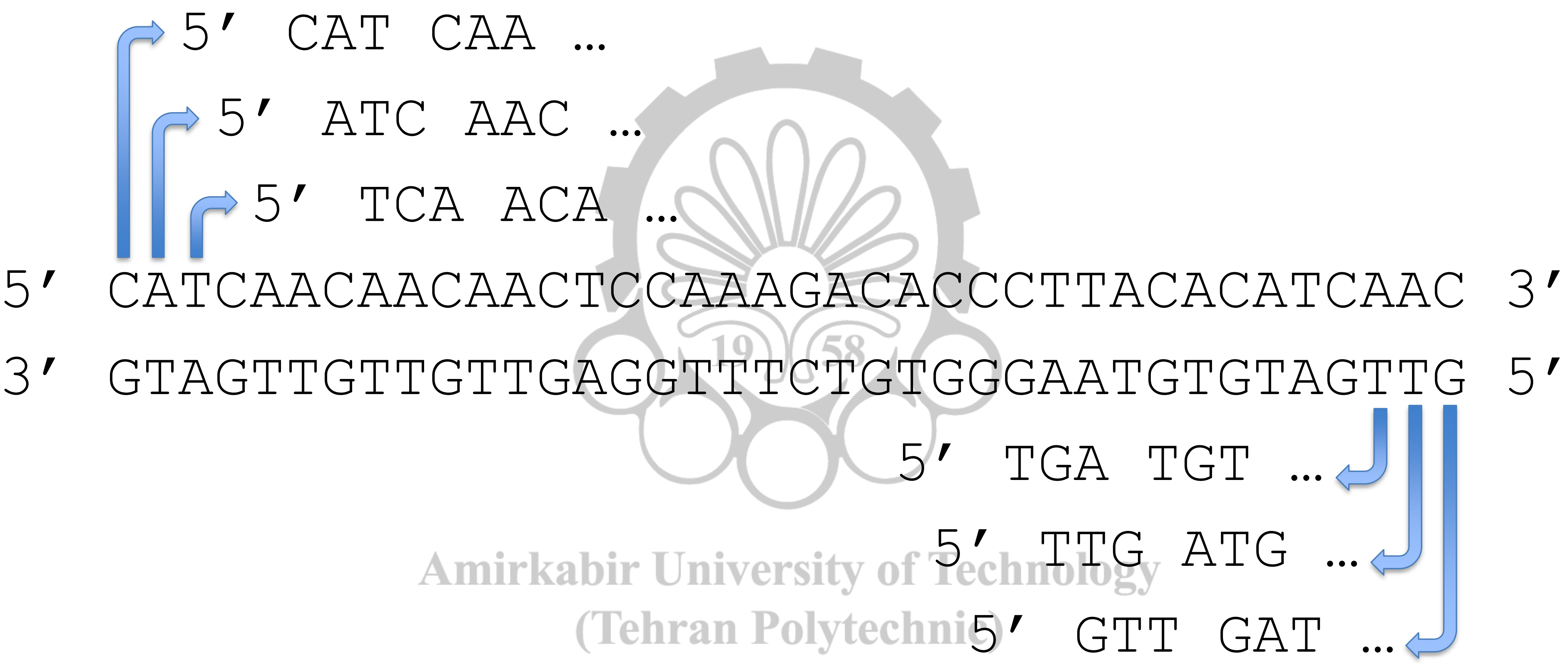
- Developed by *Stephen Altschul* at NCBI in 1990.

- Word length?
  - Typically: 3 aa for protein sequence
    11 nt for DNA sequence

- Substitution matrix?
  - Default is BLOSUM62
  - Can change under Algorithm Parameters
  - Can choose other BLOSUM or PAM matrices
  - Change other parameters here, too

- Stop-Extension Threshold?
  - Typically: 22 for proteins
    20 for DNA

DNA potentially can encode 6 protein frames



5′ CAT CAA …

5′ ATC AAC …

5′ TCA ACA …

5′ CATCAACAACAACTCCAAAGACACCCTTACACATCAAC 3′

3′ GTAGTTGTTGTTGAGGTTTCTGTGGGAATGTGTAGTTG 5′

5′ TGA TGT …

5′ TTG ATG …

5′ GTT GAT …

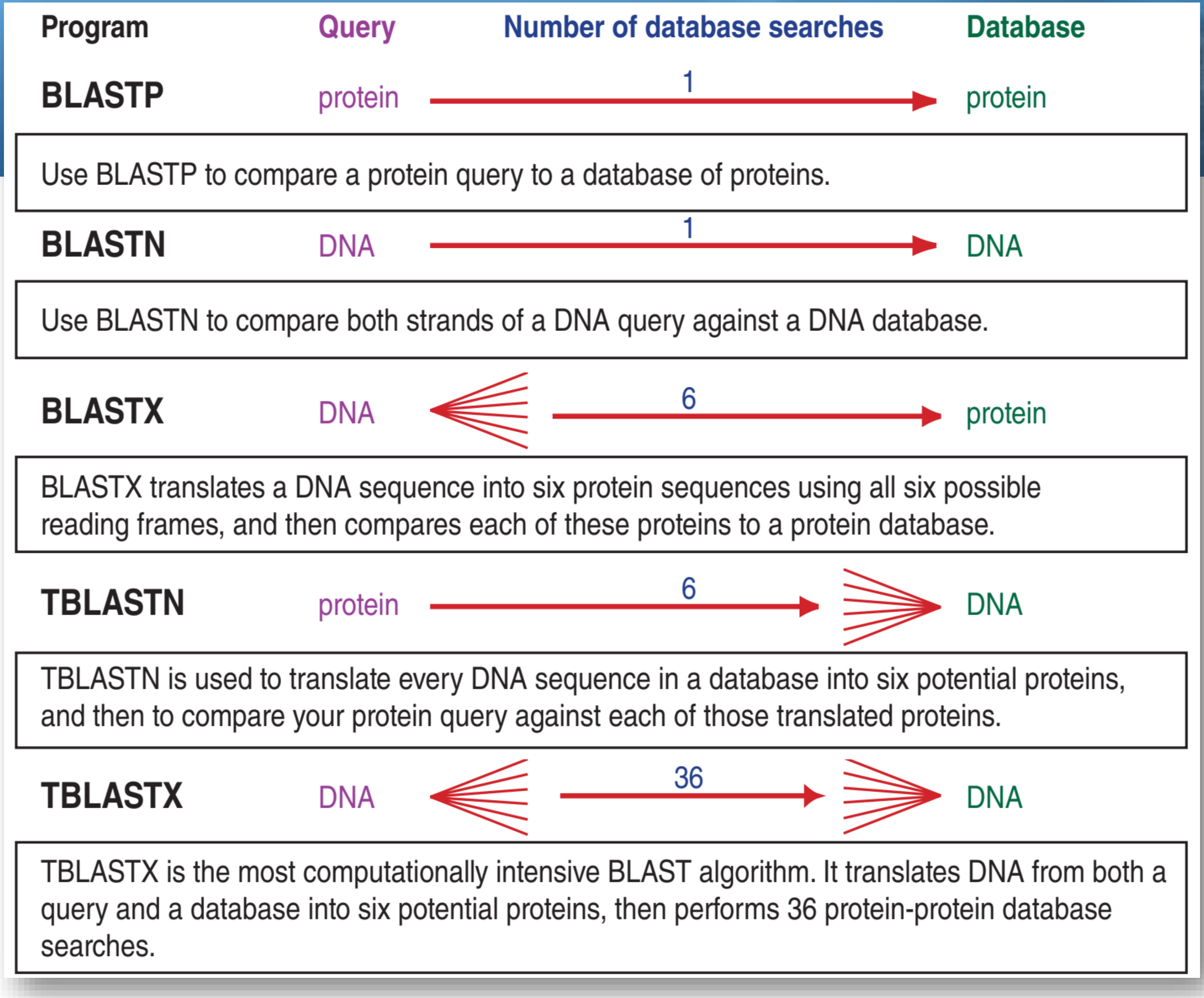
BLAST - a Family of Programs

- **BLASTN** – nucleotide (nt) sequence query against a nucleotide sequence DB (GenBank)
- **BLASTP -** protein sequence query against protein DB
- **BLASTX** – translates nt seq to six translated protein seq as query against protein DB
- **TBLASTN -** protein query against 6 translated protein from translation
- **TBLASTX -** 6-frame DNA query to 6-frame DNA translation
- **PSI-BLAST -** protein *"profile"* query against protein DB
- **PHI-BLAST -** protein *pattern* against protein DB
- ***Newest: MEGA-BLAST*** - *optimized for highly similar sequences*

Which tool should you use?

[***https://blast.ncbi.nlm.nih.gov/Blast.cgi***](https://blast.ncbi.nlm.nih.gov/Blast.cgi)

[***ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf***](ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf)



Program	Query	Number of database searches	Database
BLASTP	protein	1	protein

Use BLASTP to compare a protein query to a database of proteins.

Program	Query	Number of database searches	Database
BLASTN	DNA	1	DNA

Use BLASTN to compare both strands of a DNA query against a DNA database.

Program	Query	Number of database searches	Database
BLASTX	DNA	6	protein

BLASTX translates a DNA sequence into six protein sequences using all six possible reading frames, and then compares each of these proteins to a protein database.

Program	Query	Number of database searches	Database
TBLASTN	protein	6	DNA

TBLASTN is used to translate every DNA sequence in a database into six potential proteins, and then to compare your protein query against each of those translated proteins.

Program	Query	Number of database searches	Database
TBLASTX	DNA	36	DNA

TBLASTX is the most computationally intensive BLAST algorithm. It translates DNA from both a query and a database into six potential proteins, then performs 36 protein-protein database searches.

BLAST - Statistical Significance?

- **E-value** (expectation value): the probability that the resulting alignments are caused by random chance.
  - E = m × n × P
  - m = total number of residues in database
  - n = number of residues in query sequence
  - P = probability that an HSP is result of random chance
  - Cons: the *E*-value is proportionally affected by the database size.
- **Bit Score** (S'): measures sequence similarity independent of query sequence length and database size and is normalized based on the raw pairwise alignment score.

BLAST - Statistical Significance?

- **Bit Score** (S'): normalized score, to account for differences in size of database (m) & sequence length(n)
  - $S' = (\lambda \times S - \ln K)/\ln 2$
  - λ = Gumble distribution constant
  - S = raw alignment score
  - K = constant associated with scoring matrix
  - It is linearly related to raw alignment score, so higher S' means alignment has higher significance

Relation with E-value:

$$E = m \times n \times 2^{-S'}$$

- **Low Complexity Masking**
  - remove repeats that confound scoring

BLAST - Statistical Significance?

- Conclusions based on E-value:
  - $E < 1e{-}50$: there should be an extremely high confidence that the database match is a result of homologous relationships.
  - $1e{-}50 < E < 0.01$: the match can be considered a result of homology.
  - $0.01 < E < 10$: the match is considered not significant, but may hint at a tentative remote homology relationship.
  - $E > 10$, the sequences under consideration are unrelated.

Detailed Steps in BLAST algorithm

1. *Remove low-complexity regions (LCRs)*
2. *Make a list* (dictionary): all words of length 3 aa or 11 nt
3. *Augment list* to include similar words
4. *Store list* in a search tree (*data structure*)
5. *Scan database* for occurrences of words in search tree
6. *Connect* nearby occurrences
7. *Extend* matches (words) in both directions
8. *Prune list* of matches using a score *threshold*
9. *Evaluate significance* of each remaining match
10. *Perform Smith-Waterman* to get alignment

1: Filter low-complexity regions (LCRs)

- Low complexity regions, transmembrane regions and coiled-coil regions often display significant similarity without homology.
- Low complexity sequences can yield false positives.
- Screen them out of your query sequences! *When appropriate!*

e.g., for GGGG:

L! = 4!=4x3x2x1= 24

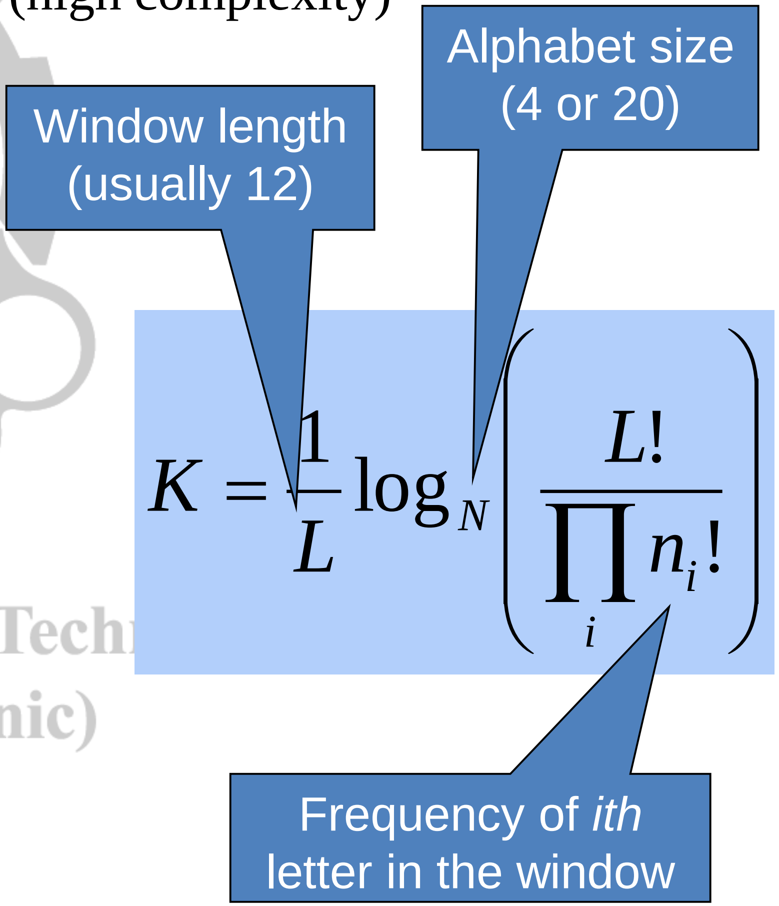
n_G=4 n_T=n_A=n_C=0

Π n_i! = 4!x0!x0!x0! = 24

K=1/4 $\log_4$ (24/24) = 0

For CGTA: K=1/4 $\log_4$(24/1) = 0.57

K = computational complexity; varies from 0 (very low complexity) to 1 (high complexity)



Alphabet size (4 or 20)

Window length (usually 12)

$$K = \frac{1}{L} \log_N \left(\frac{L!}{\prod_i n_i!} \right)$$

Frequency of *ith* letter in the window

2: List all words in query

YGGFMTSEKSQTPLVTLFKNAIIKNAHKKGQ

YGG

GGF

GFM

FMT

MTS

TSE

SEK

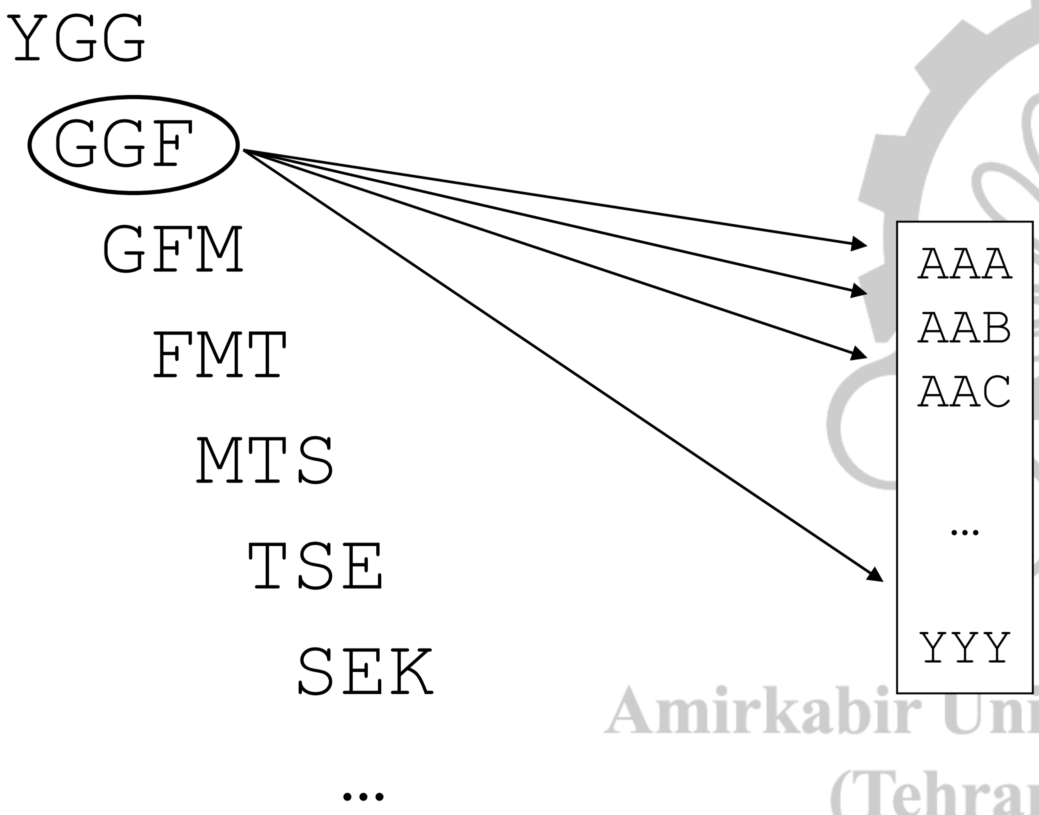
…

$$WordCount = Len_{query} - Len_{word} + 1$$

$$WordCount = 31 - 3 + 1 = 29$$

3: Augment word list

YGGFMTSEKSQTPLVTLFKNAIIKNAHKKGQ



YGG

GGF

GFM

FMT

MTS

TSE

SEK

…

AAA

AAB

AAC

…

YYY

$20^3 = 8000$
possible matches

3: Augment word list (Cont.)

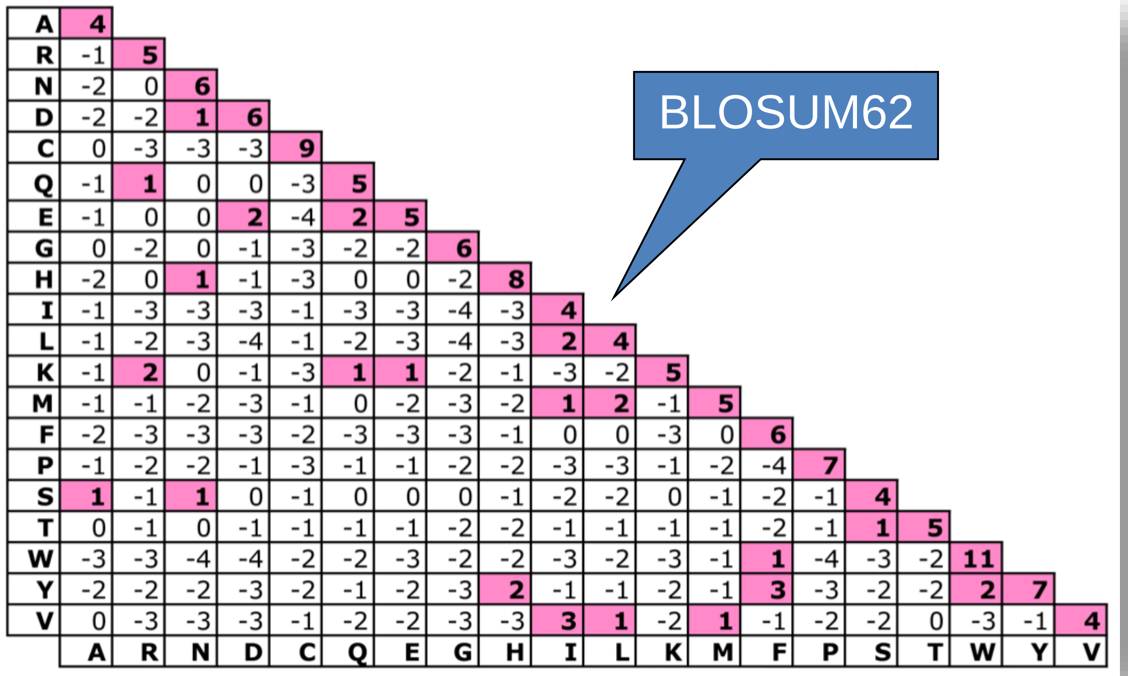
BLOSUM62 scores

```
G     G     F
A     A     A
-------------
0  +  0  + -2  =  -2
```

Non-match

```
G     G     F
G     G     Y
-------------
6  +  6  +  3  =  15
```

Match



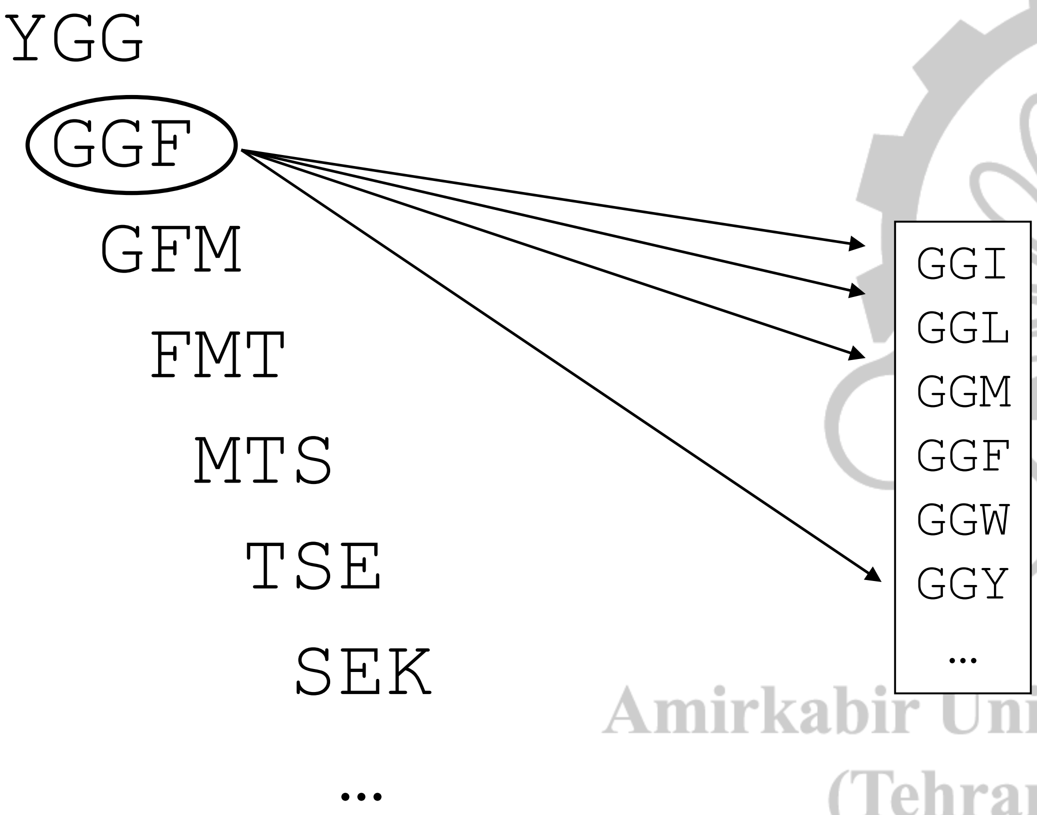
BLOSUM62

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

A user-specified **threshold, T**, determines which 3-letter words are considered matches and non-matches

3: Augment word list (Cont.)

YGGFMTSEKSQTPLVTLFKNAIIKNAHKKGQ



YGG

GGF

GFM

FMT

MTS

TSE

SEK

…

GGI

GGL

GGM

GGF

GGW

GGY

…

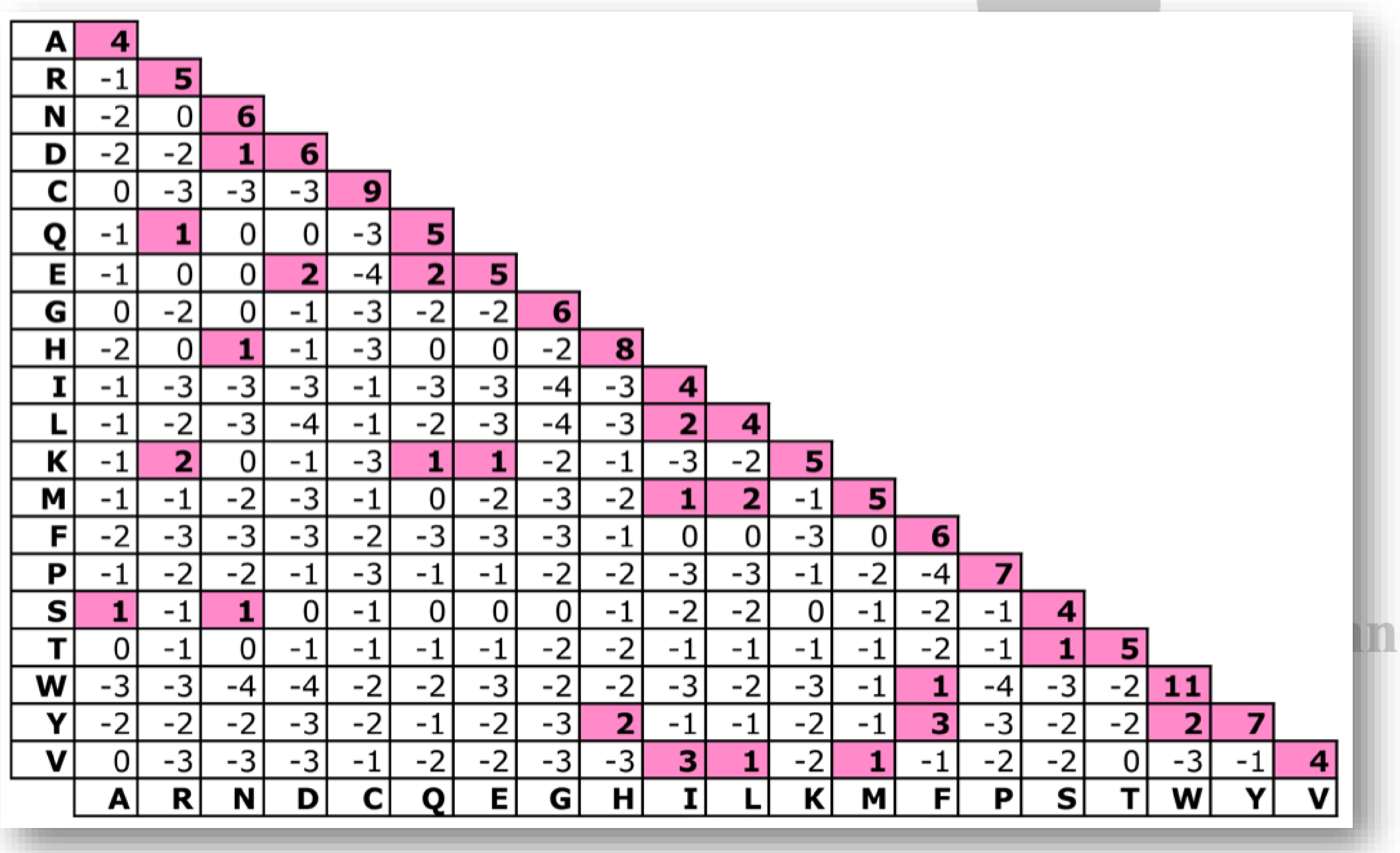
Observation:

Selecting only words with score > T greatly reduces number of possible matches

otherwise, 20^3 for 3-letter words from amino acid sequences!

Example

Find all words that match EAM with a score greater than or equal to 11

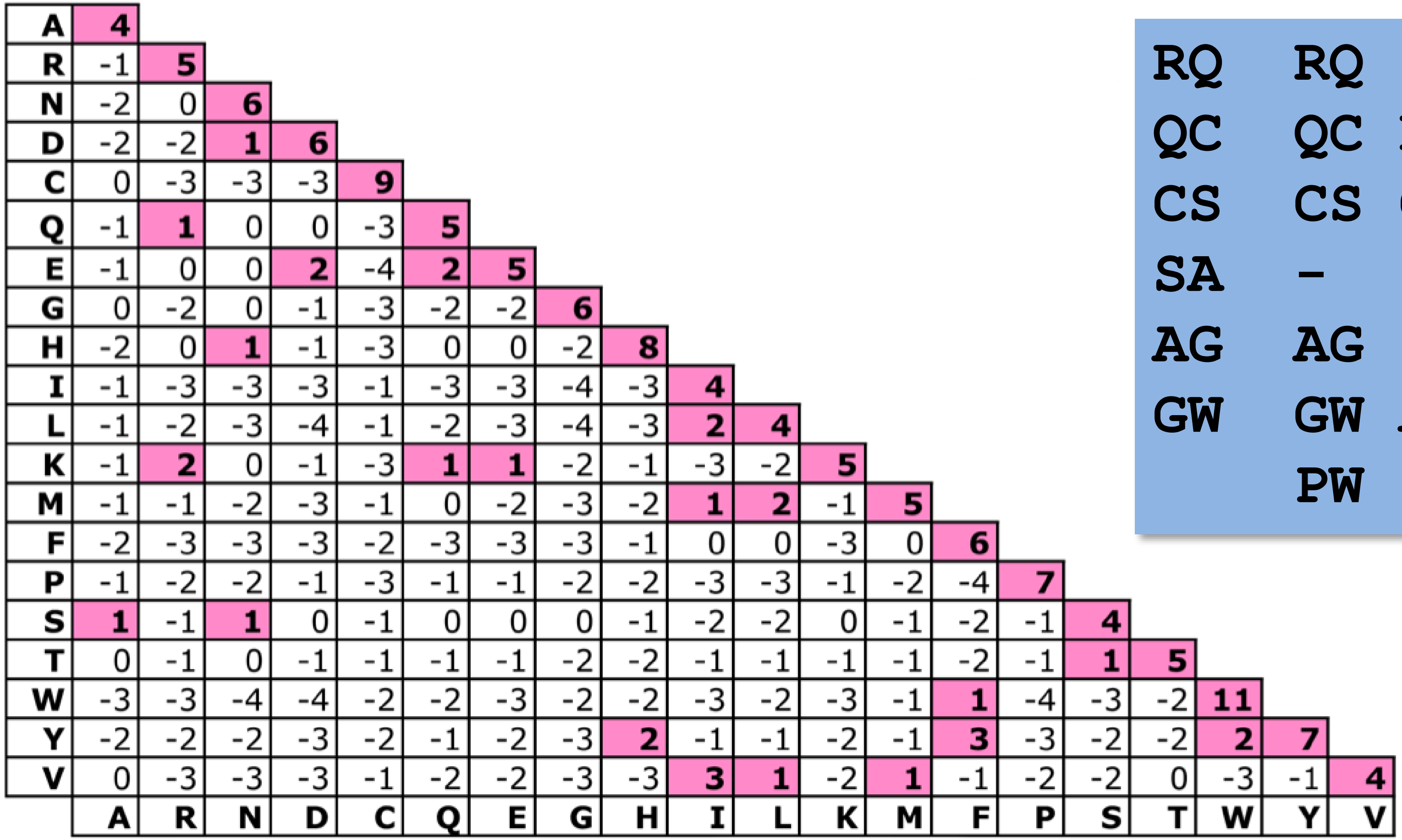


A	**4**																			
R	-1	**5**																		
N	-2	0	**6**																	
D	-2	-2	**1**	**6**																
C	0	-3	-3	-3	**9**															
Q	-1	**1**	0	0	-3	**5**														
E	-1	0	0	**2**	-4	**2**	**5**													
G	0	-2	0	-1	-3	-2	-2	**6**												
H	-2	0	**1**	-1	-3	0	0	-2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

```
EAM 5 + 4 + 5 = 14
DAM 2 + 4 + 5 = 11
QAM 2 + 4 + 5 = 11
ESM 5 + 1 + 5 = 11
EAL 5 + 4 + 2 = 11
```

Example 2

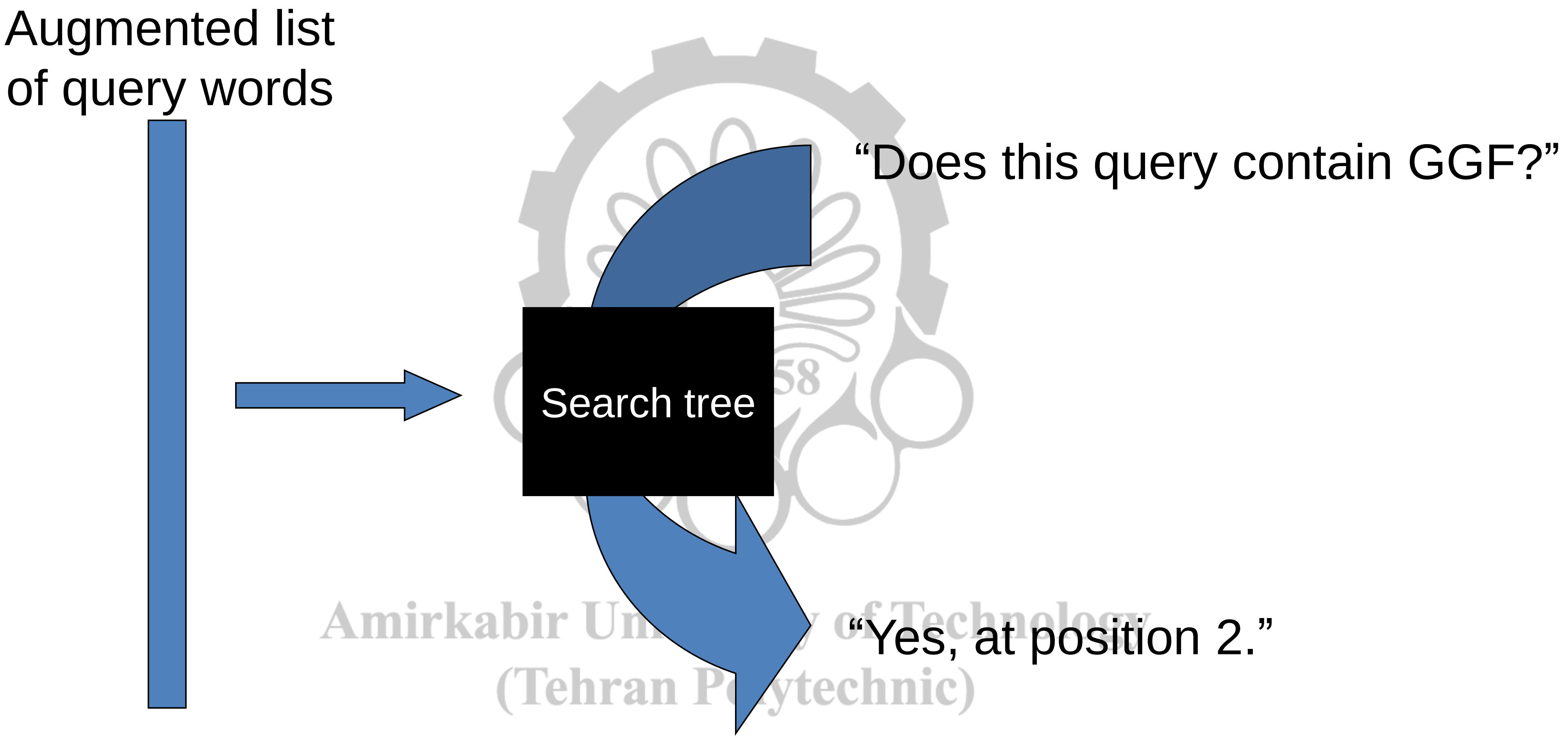
Find all words with size 2 and score greater than 8 for **RQCSAGW**



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

RQ	**RQ**
QC	**QC RC EC NC DC KC MC SC**
CS	**CS CA CN CD CQ CE CG CK CT**
SA	**–**
AG	**AG**
GW	**GW AW RW NW DW QW EW HW KW**
	PW SW TW WW

4: Store words in search tree



Search Tree (Trie)

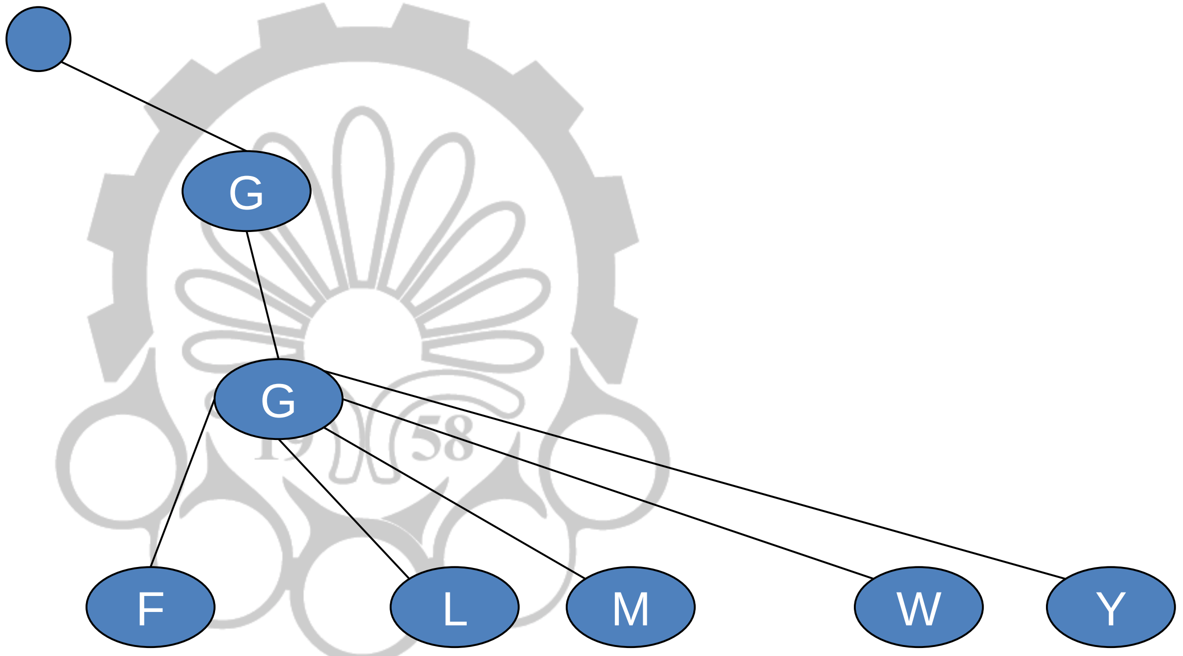
GGF

GGL

GGM

GGW

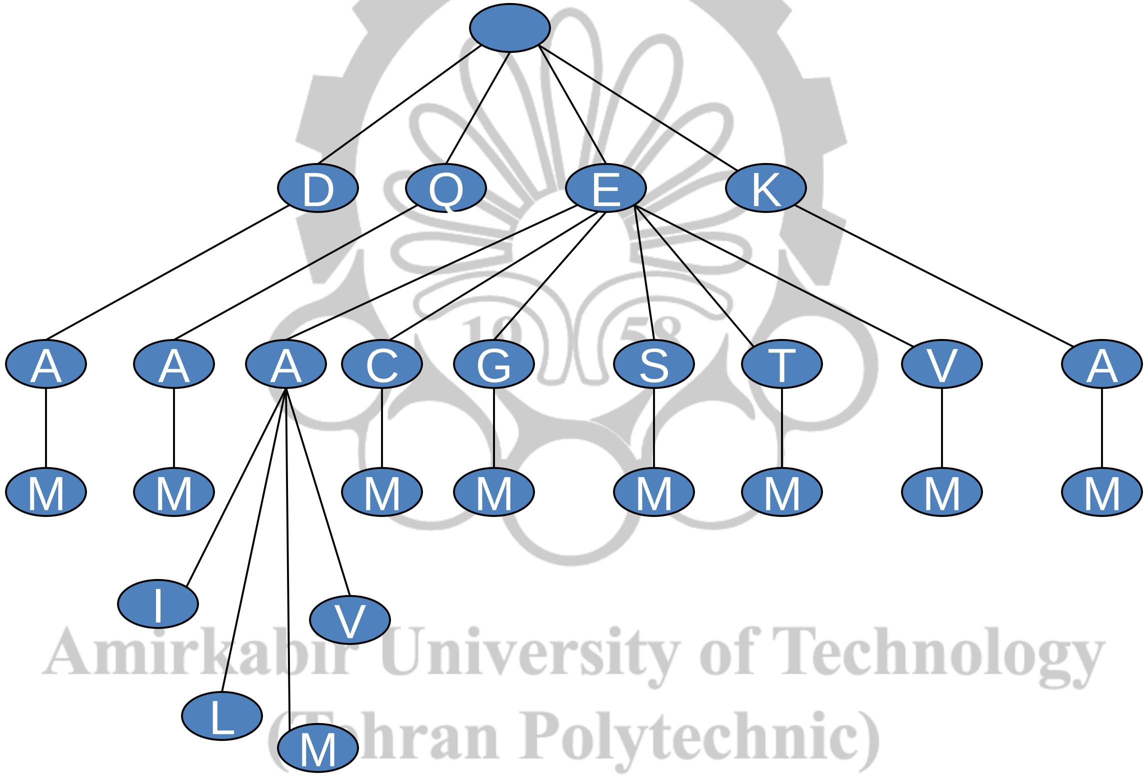
GGY



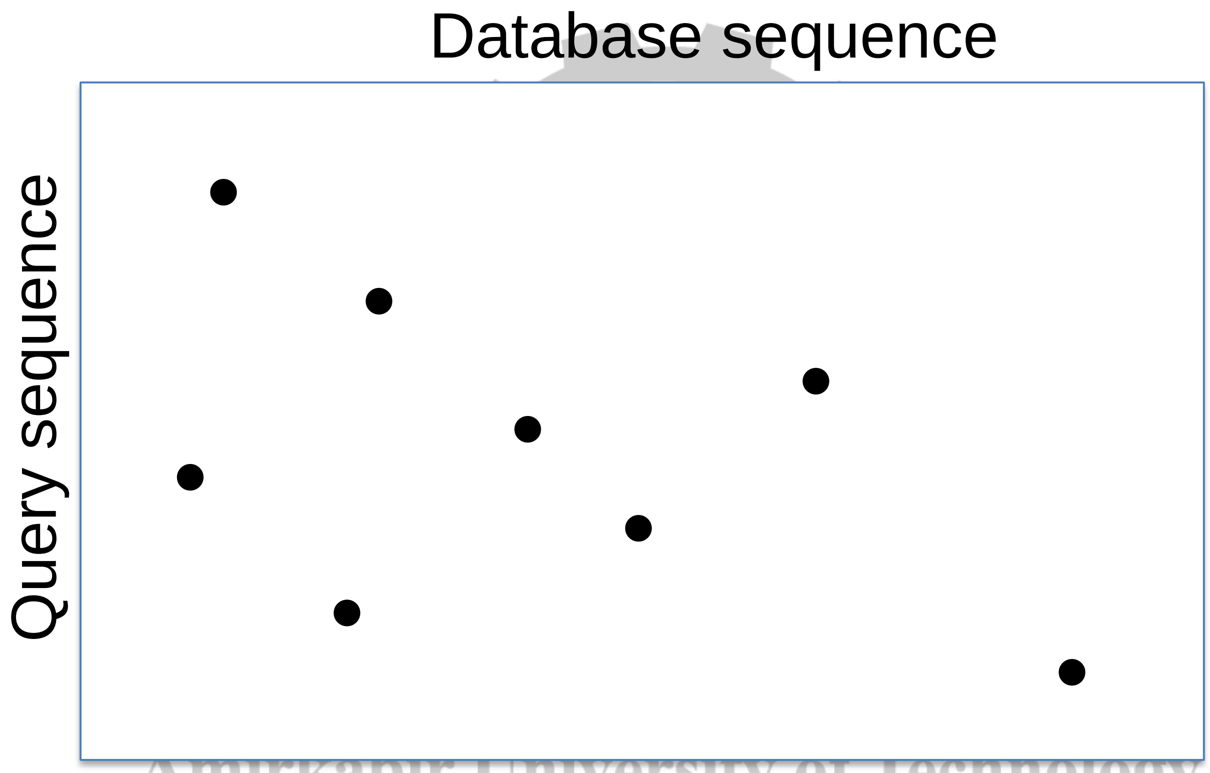
Trie Example

Put this word list into a search tree

DAM
QAM
EAM
KAM
ECM
EGM
ESM
ETM
EVM
EAI
EAL
EAV

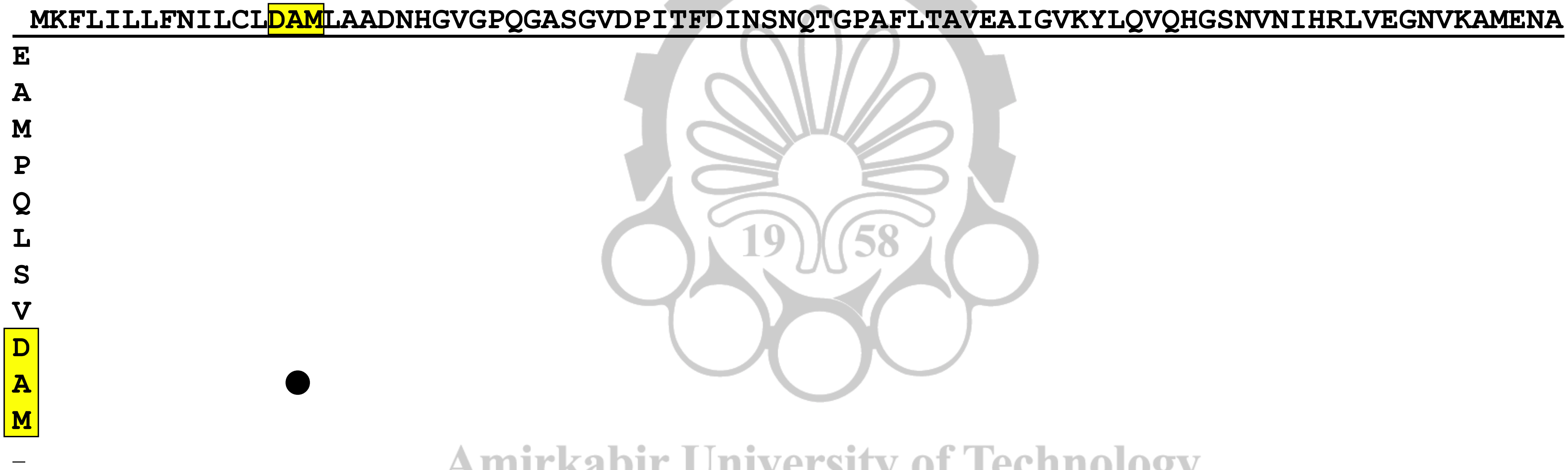


5: Scan the database sequences



Example

Scan this "database" for occurrences of your words

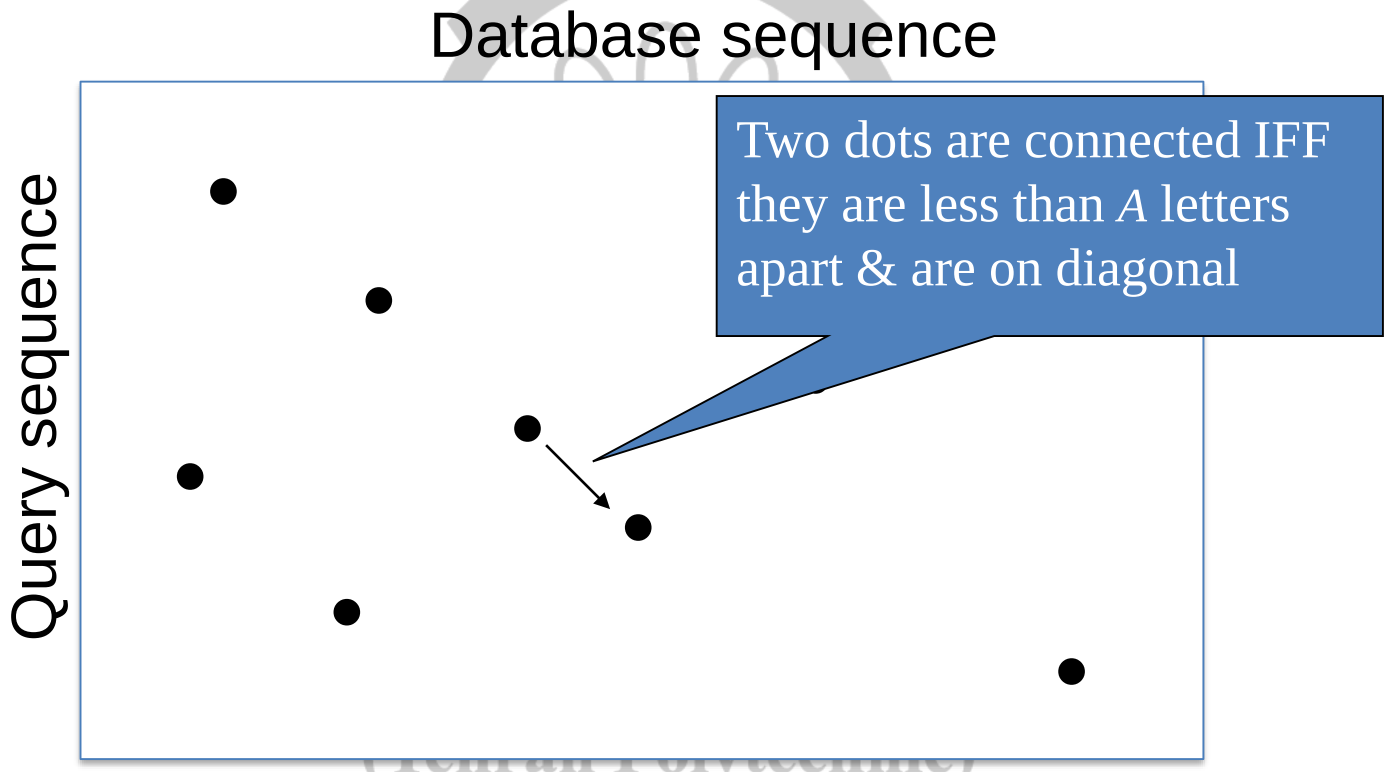


MKFLILLFNILCLDAMLAADNHGVGPQGASGVDPITFDINSNQTGPAFLTAVEAIGVKYLQVQHGSNVNIHRLVEGNVKAMENA

E
A
M
P
Q
L
S
V
D
A
M
_

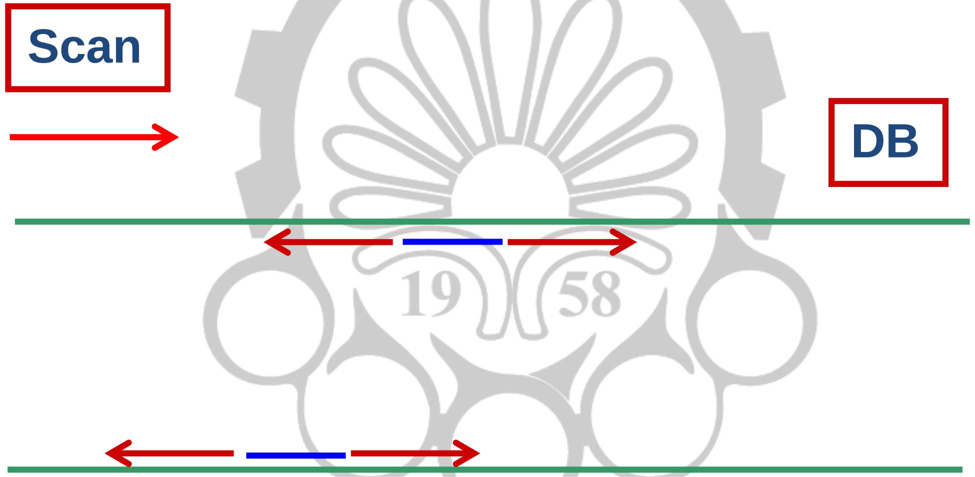
6: Connect nearby occurrences

- (diagonal matches in Gapped BLAST)



Two dots are connected IFF they are less than *A* letters apart & are on diagonal

7: Extend matches in both directions



7: Extend matches, calculating score at each step

```
L P  P Q G  L L  Query sequence
M P  P E G  L L  Database sequence
     <word>
     7 2 6       BLOSUM62 scores
                 word score = 15
<---       --->
2 7  7 2 6  4 4  HSP SCORE = 32
         (High Scoring Pair)
```

- Each match is extended to left & right until a negative BLOSUM62 score is encountered
- Extension step typically accounts for > 90% of execution time

8&9: Prune matches & Evaluate significance

- Prune matches:
  - Discard all matches that score below defined threshold

- Evaluate significance:
  - BLAST uses an analytical statistical significance calculation

10: Use SW algorithm to generate alignment

- *ONLY* significant matches are re-analyzed using Smith-Waterman DP algorithm.
- Alignments reported by BLAST are produced by dynamic programming

BLAST: What is a "Hit"?

- A ***hit*** is a *w*-length word in database that aligns with a word from query sequence with score > *T*
- BLAST looks for ***hits*** instead of exact matches
  - Allows word size to be kept larger for speed, without sacrificing sensitivity
- Typically:
  - *w* = 3-5 for amino acids, *w* = 11-12 for DNA
- *T* is the most critical parameter:
  - ↑*T* ⇒ ↓ “background” hits (faster)
  - ↓*T* ⇒ ↑ ability to detect more distant relationships (at cost of increased noise)

Tips for BLAST Similarity Searches

- If you don’t know, use default parameters first
- Try several programs & several parameter settings
- If possible, search on *protein* sequence level

- **Scoring matrices:**
  - PAM1 / BLOSUM80: if expect/want less divergent proteins
  - PAM120 / BLOSUM62: "average" proteins
  - PAM250 / BLOSUM45: if need to find more divergent proteins

- **Proteins:**

  >25-30% identity (and >100aa) -> likely related
  15-25% identity -> twilight zone
  <15% identity -> likely unrelated

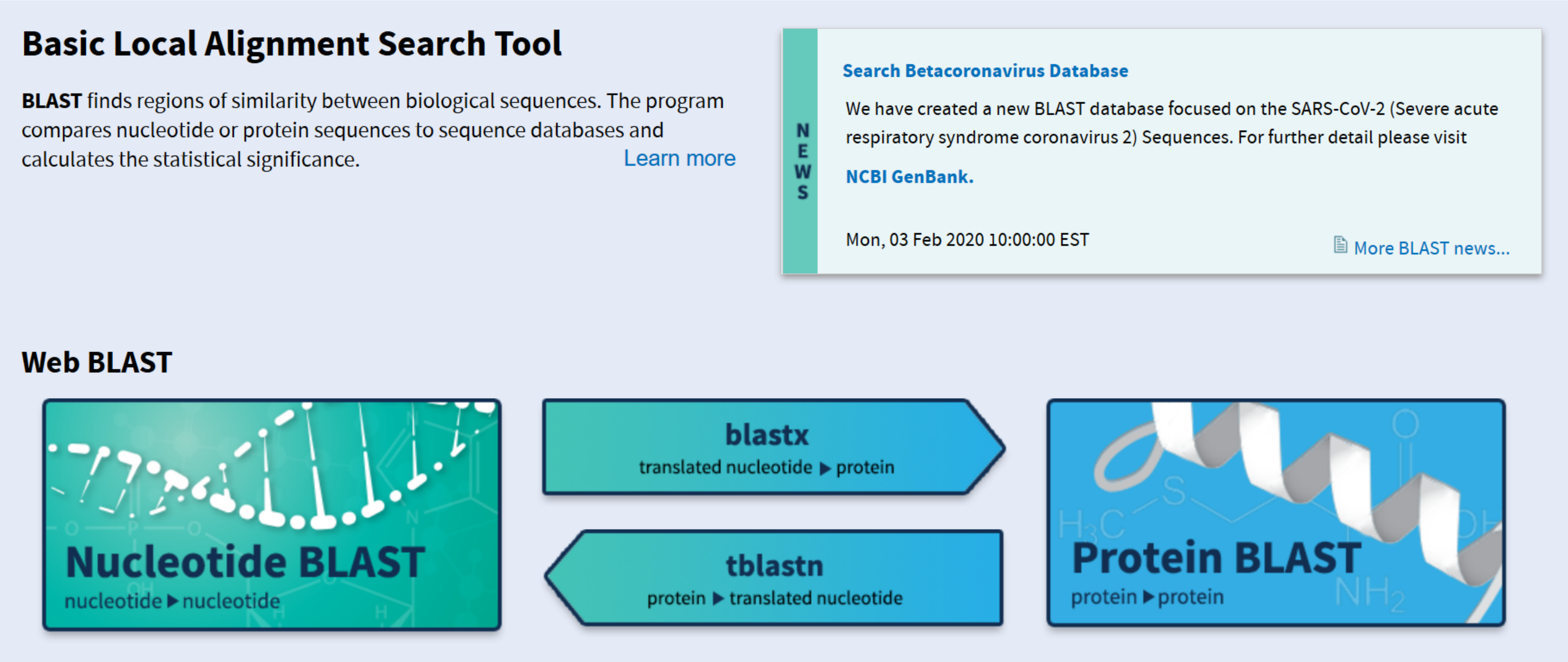
Practical Issues

- Searching on DNA or protein level?
- In general, protein - encoding DNA should be translated!

- DNA yields more random matches:
  - 25% for DNA vs. 5% for proteins
- DNA databases are larger and grow faster

- Selection (*generally*) acts on protein level
  - *Synonymous mutations* are *usually* neutral
  - DNA sequence similarity decays faster

NCBI: BLAST



Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. Learn more

NEWS

Search Betacoronavirus Database

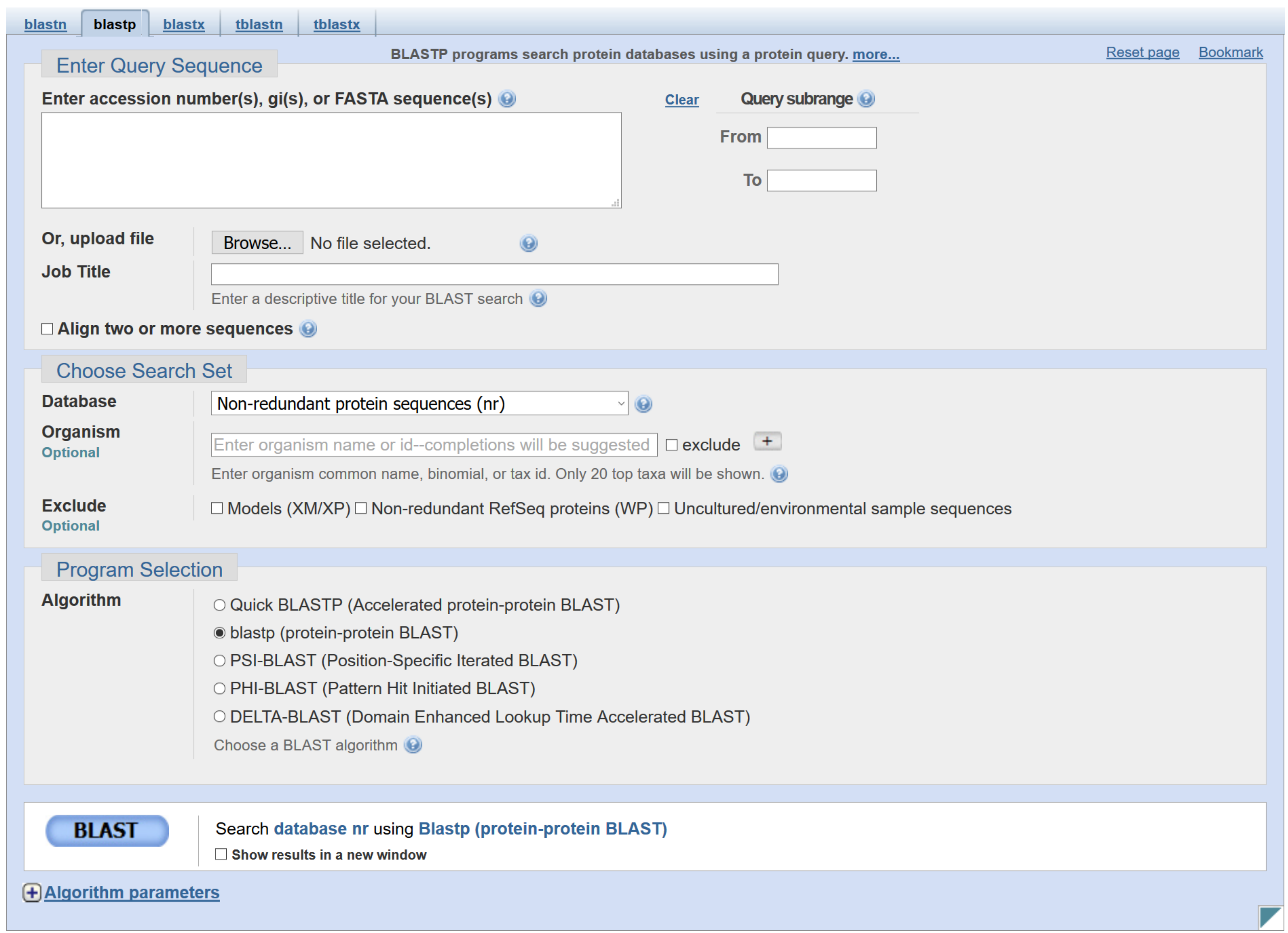
We have created a new BLAST database focused on the SARS-CoV-2 (Severe acute respiratory syndrome coronavirus 2) Sequences. For further detail please visit **NCBI GenBank.**

Mon, 03 Feb 2020 10:00:00 EST

More BLAST news...

Web BLAST

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



blastn | **blastp** | blastx | tblastn | tblastx

BLASTP programs search protein databases using a protein query. more...

Reset page Bookmark

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

Clear

Query subrange

From

To

Or, upload file Browse... No file selected.

Job Title

Enter a descriptive title for your BLAST search

□ **Align two or more sequences**

Choose Search Set

Database Non-redundant protein sequences (nr)

Organism
Optional

Enter organism name or id--completions will be suggested □ exclude +

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.

Exclude
Optional

□ Models (XM/XP) □ Non-redundant RefSeq proteins (WP) □ Uncultured/environmental sample sequences

Program Selection

Algorithm

○ Quick BLASTP (Accelerated protein-protein BLAST)
● blastp (protein-protein BLAST)
○ PSI-BLAST (Position-Specific Iterated BLAST)
○ PHI-BLAST (Pattern Hit Initiated BLAST)
○ DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)

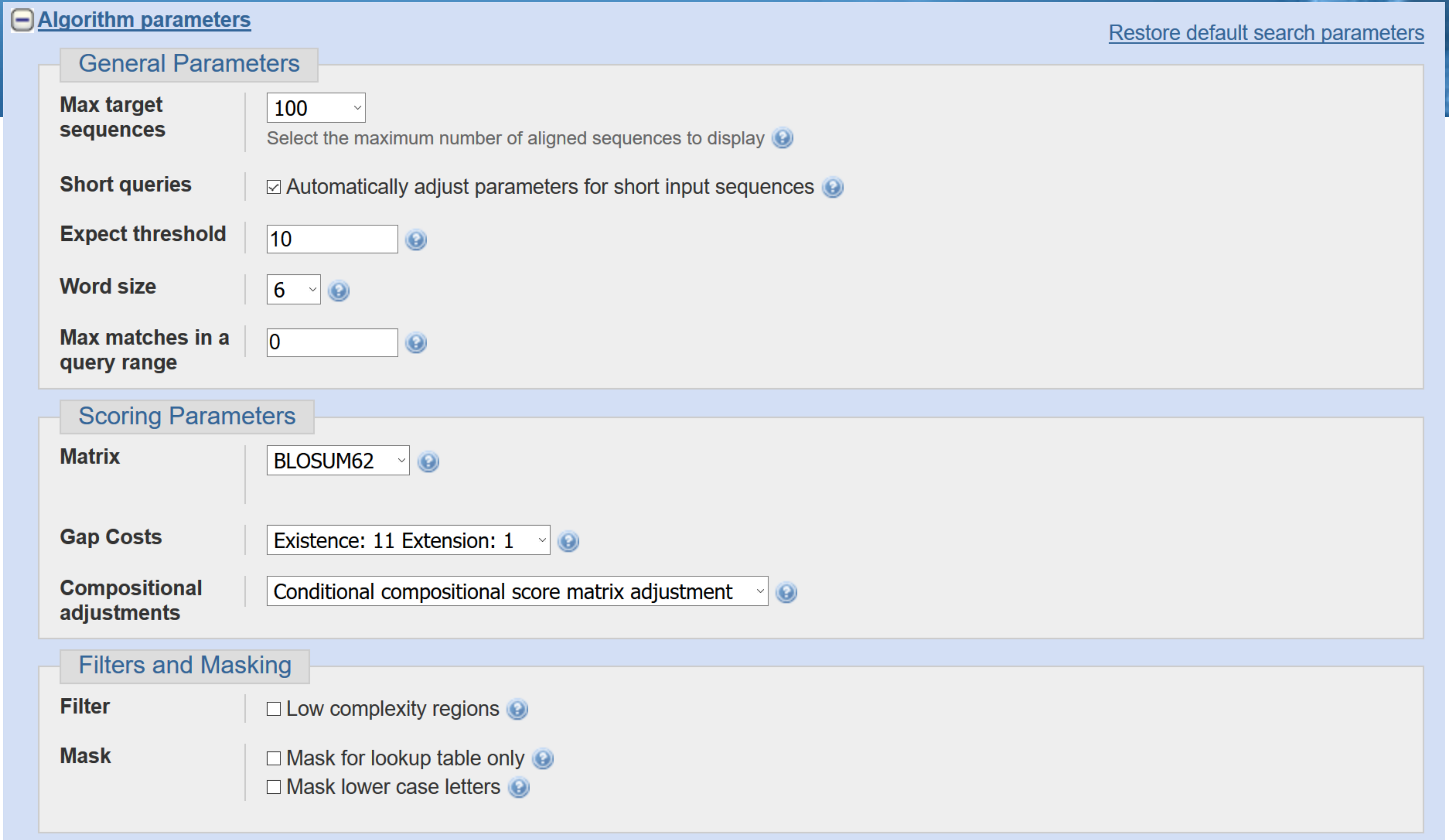
Choose a BLAST algorithm

BLAST

Search **database nr** using **Blastp (protein-protein BLAST)**

□ **Show results in a new window**

+ Algorithm parameters



⊖ Algorithm parameters

Restore default search parameters

General Parameters

Max target sequences	100 Select the maximum number of aligned sequences to display
Short queries	☑ Automatically adjust parameters for short input sequences
Expect threshold	10
Word size	6
Max matches in a query range	0

Scoring Parameters

Matrix	BLOSUM62
Gap Costs	Existence: 11 Extension: 1
Compositional adjustments	Conditional compositional score matrix adjustment

Filters and Masking

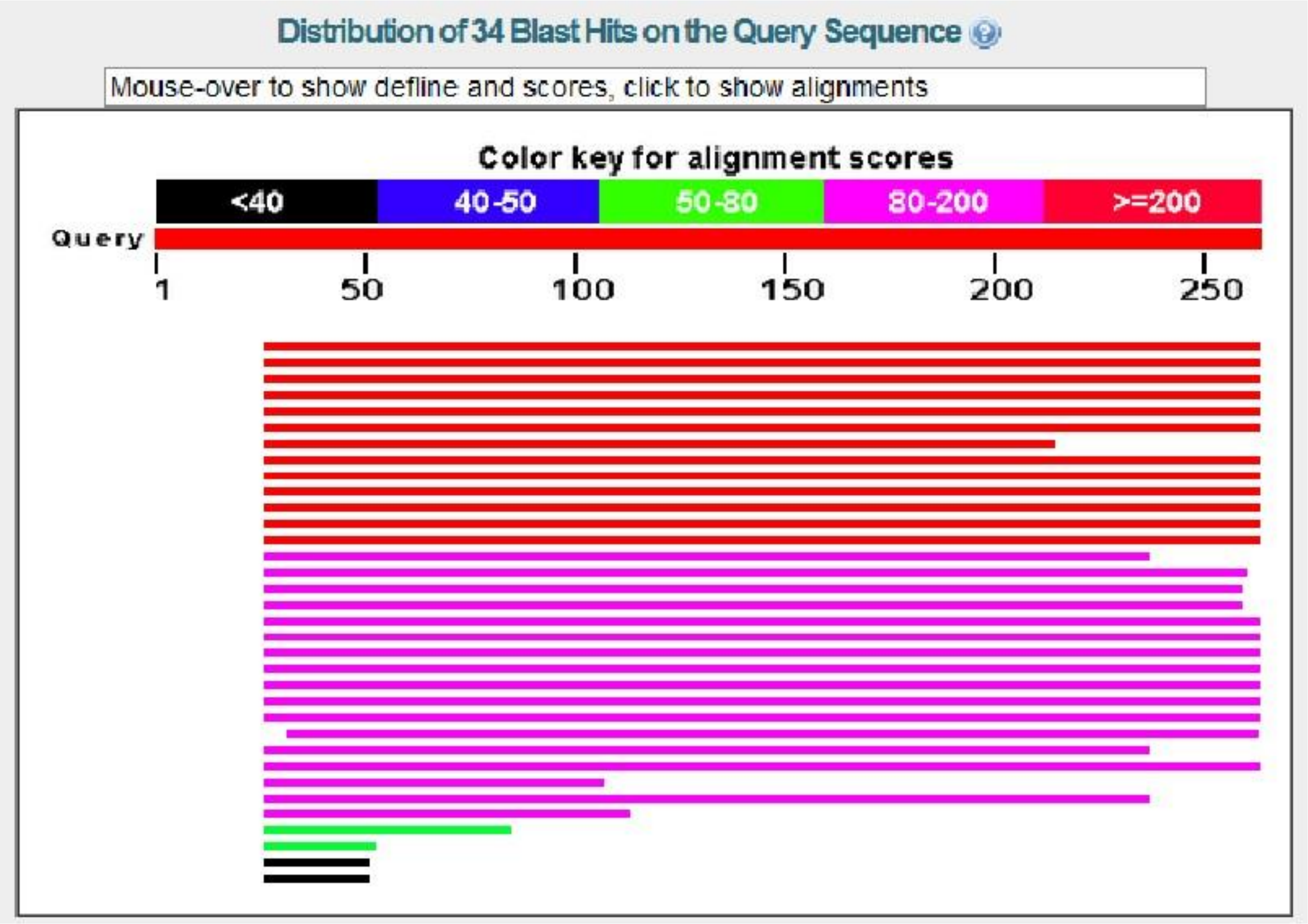
Filter	□ Low complexity regions
Mask	□ Mask for lookup table only □ Mask lower case letters

Example: P01308 (INS_HUMAN)

- Sequence:

```
MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLV
CGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEG
SLQKRGIVEQCCTSICSLYQLENYCN
```

BLAST Output



Distribution of 34 Blast Hits on the Query Sequence

Mouse-over to show defline and scores, click to show alignments

FASTA (FAST ALL)

FASTA

- FASTA was the first database similarity search tool.
- It uses a **hashing** strategy to find matches for a short stretch of identical residues with a length of *k*.
- The string of residues is known as *ktuples* or *ktups*, which are equivalent to words in BLAST, but are normally shorter.
  - A ktup is composed of 2 residues for protein sequences and 6 residues for DNA sequences.

https://www.ebi.ac.uk/Tools/sss/fasta/

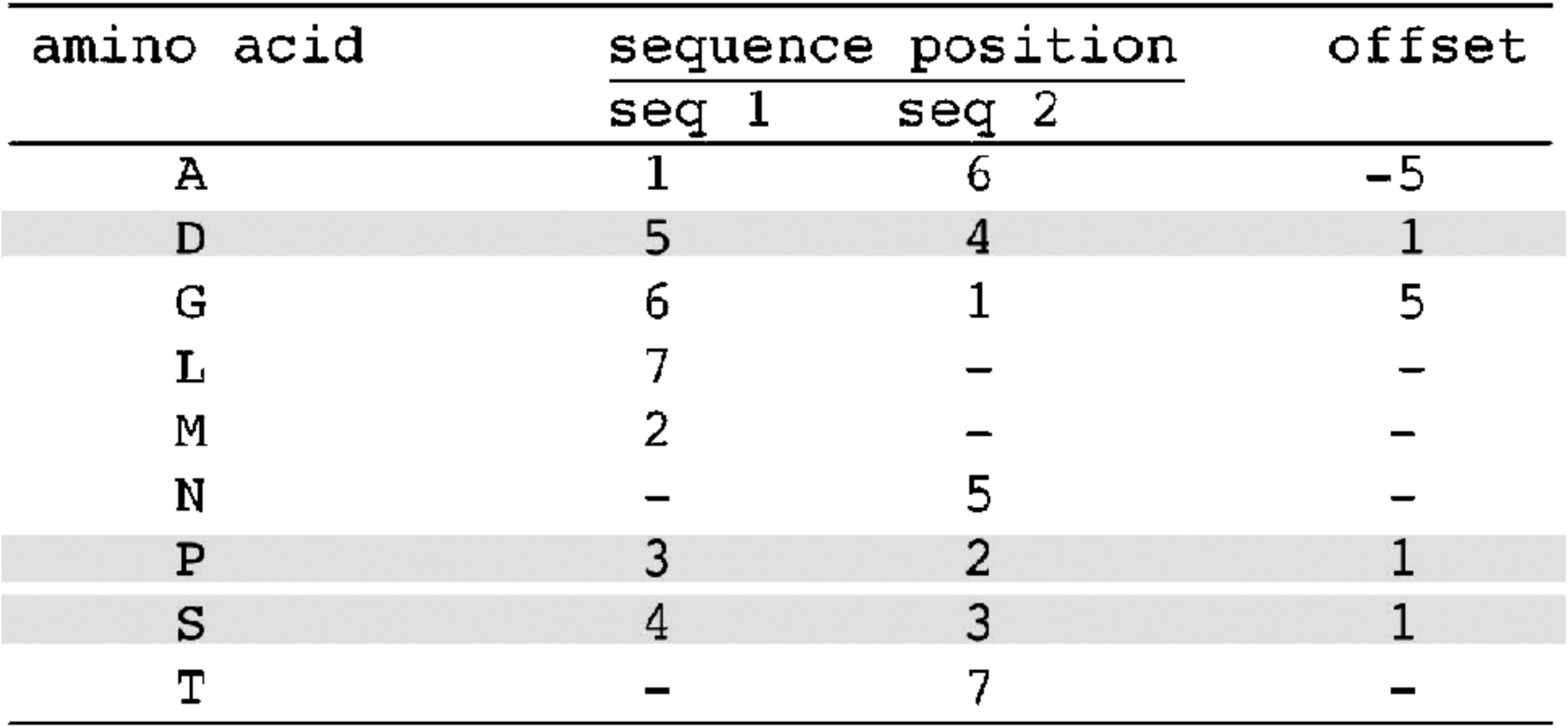
Steps in FASTA

- Step 1 : identify ktups between two sequences by using the hashing strategy.
- Step 2 : narrow down the high similarity regions between the two sequences.
- Step 3 : the gapped alignment is refined further using the Smith–Waterman algorithm to produce a final alignment.
- Step 4 : perform a statistical evaluation of the final alignment as in BLAST, which produces the *E*-value.

Step 1: Construct a Hashing Table

```
Seq1 = AMPSDGL
Seq2 = GPSDNAT
```

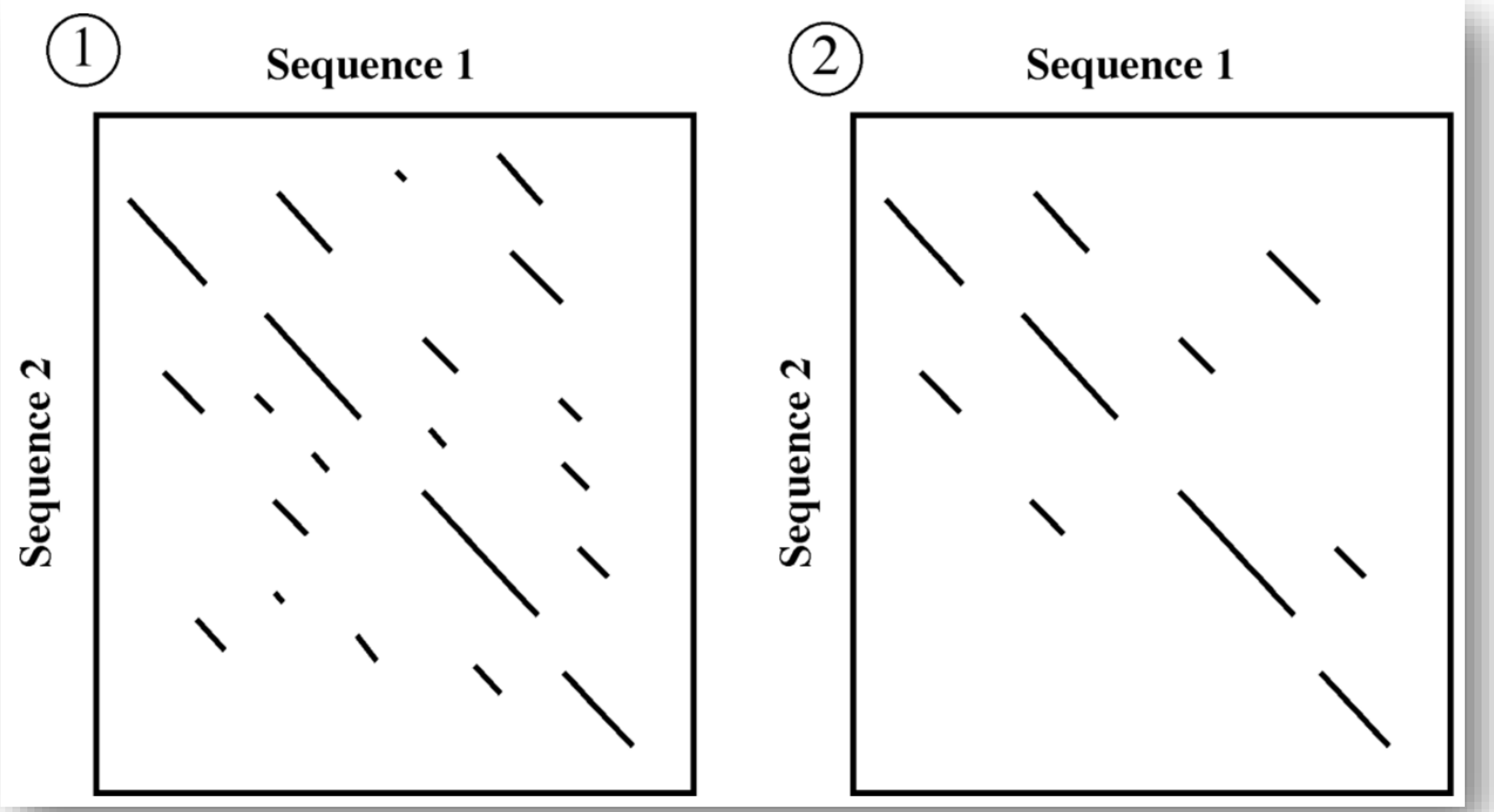
```
AMPSDGL-
  |||
-GPSDNAT
```



amino acid	sequence position		offset
	seq 1	seq 2	
A	1	6	-5
D	5	4	1
G	6	1	5
L	7	–	–
M	2	–	–
N	–	5	–
P	3	2	1
S	4	3	1
T	-	7	-

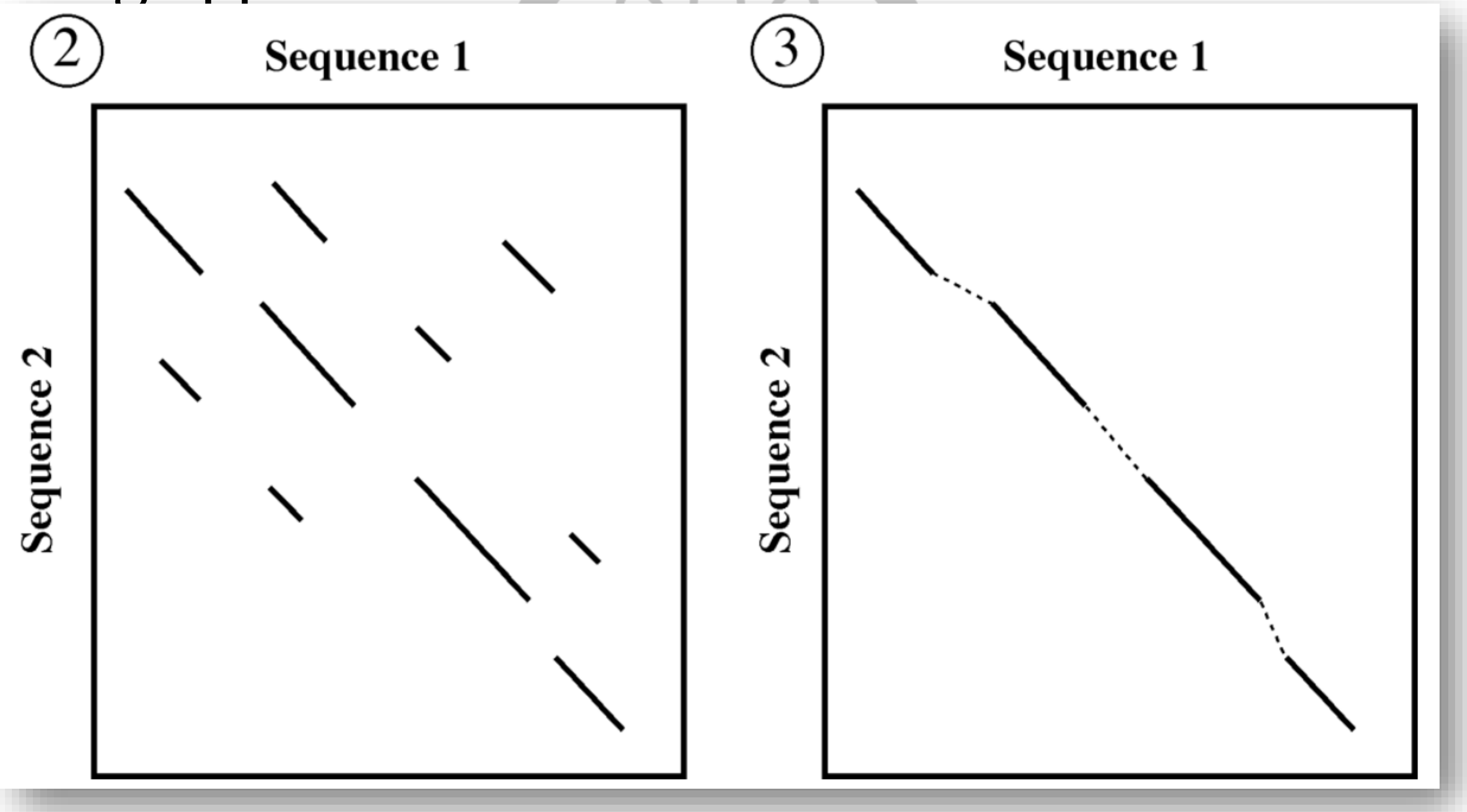
Step 2: Narrow Down the Similarity Regions

- The alignments are scored according to a particular scoring matrix. Only the ten best alignments are selected.



Step 3: Refined the Gapped Alignment

- The alignments in the same diagonal are selected and joined to form a single gapped alignment, which is optimized using the dynamic programming approach.



Step 4: Perform a Statistical Evaluation

- FASTA also uses *E*-values and bit scores.
- Estimation of the two parameters in FASTA is essentially the same as in BLAST.
- In addition, the FASTA output provides one more statistical parameter, the *Z*-score.
  - *Z*-score describes the number of standard deviations from the mean score for the database search.
- The higher *Z*-score means the more significant match.
  - *Z*-score > 15: extremely significant with certainty of a homologous relationship.
  - *5* < *Z*-score < 15: sequence pair can be described as highly probable homologs.
  - *Z* < 5: relationship is described as less certain.

BLAST vs FASTA

- **Seeding**:
  - – BLAST integrates scoring matrix into first phase
  - – FASTA requires exact matches (uses hashing)

- FASTA uses shorter word sizes - so it gives more sensitive results with a better coverage rate for homologs.
- BLAST increases search speed by finding fewer, but better, words during initial screening phase.

- **Results**:
  - – BLAST can return multiple best scoring alignments
  - – FASTA returns only one final alignment

BLAST Notes - & DP Alternatives

- BLAST uses heuristics: it may miss some good matches
  - It has been estimated that for some families of protein sequences BLAST can miss 30% of truly significant matches.
- But, it’s fast: 50 - 100X faster than Smith-Waterman (SW) DP
- Large impact:
  - NCBI’s BLAST server handles more than 100,000 queries/day
  - Most used bioinformatics program in the world!
- Increased availability of parallel processing has made DP-based approaches feasible: 2 DP-based web servers: both more sensitive than BLAST
  - **Scan Protein Sequence**: http://www.ebi.ac.uk/scanps/index.html
    Implements modified SW optimized for parallel processing
  - **ParAlign**: www.paralign.org - parallel SW or heuristics

References

- Mostly used:
  - – Essential bioinformatics, Chapter 3 (Database Similarity Searching)
- Second reference:
  - – Bioinformatics and functional genomics, Chapter 3 (Basic Local Alignment Search Tool (BLAST))
- IP notice: some slides were selected from Drena Dobbs’ slides.

Thanks for your attention