Heuristic Methods for Sequence Database Searching

BMI/CS 576

www.biostat.wisc.edu/bmi576/

Irene Ong

irene.ong@wisc.edu

Fall 2017

Heuristic Alignment Algorithms

- So far, all the alignment algorithms are guaranteed to find the optimal score according to the specified scoring scheme
- Affine gap alignment algorithm is the most sensitive sequence matching method available
- However, these alignment methods are slow
- If we want to search through many sequences, time becomes an important issue

Database Search

- Sequences:
  - Nucleotides: A,C,G,T
  - Amino-acids: A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V
- Database search:
  - Database:

    ANIKMCQSTYR….

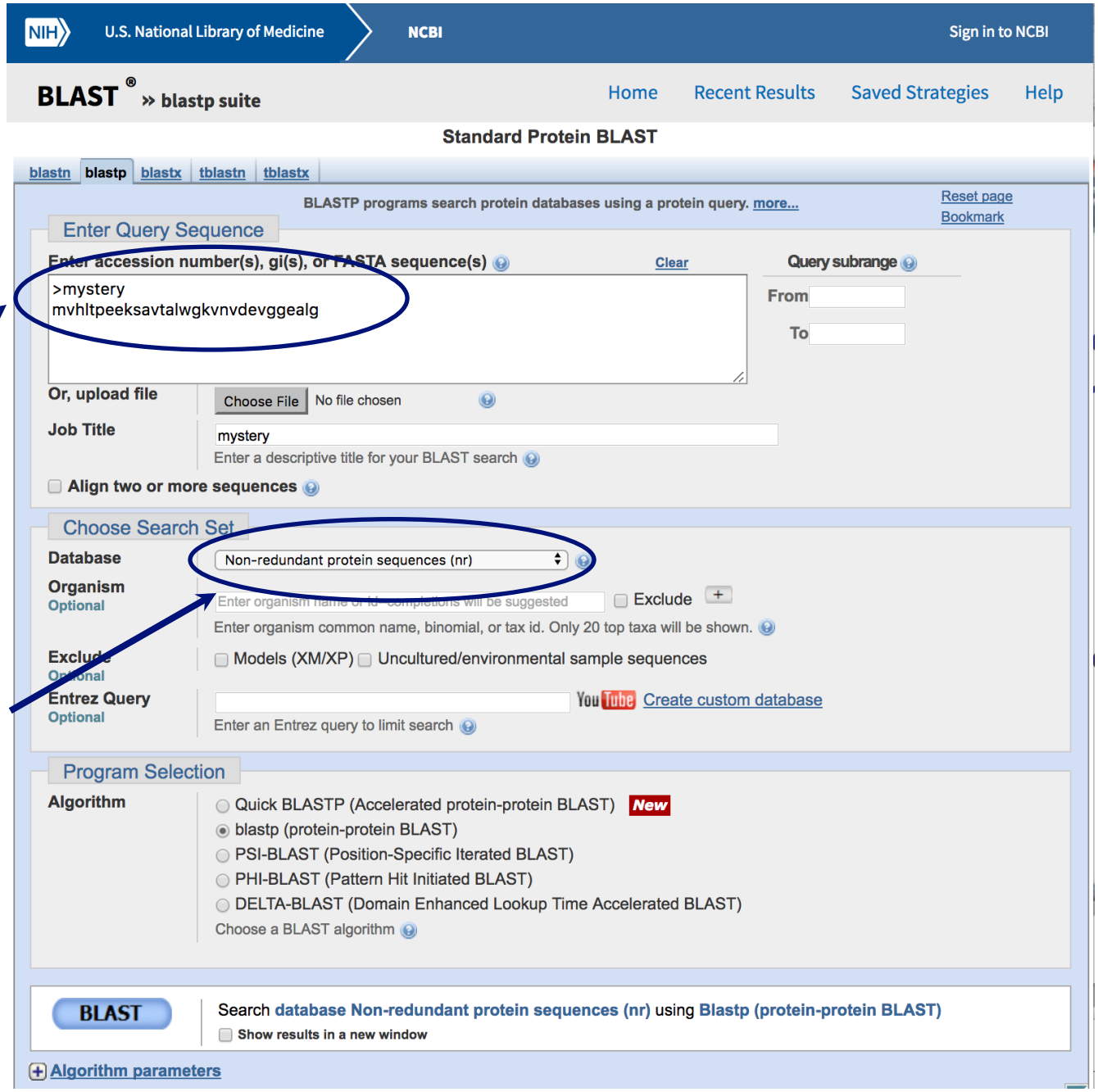
    GFNDIKHWLFSV….

    EGQMPSYVKGA….

    ……
  - Query: AGIKM
  - Output: sequences similar to query

How do we measure similarity?

- Count the number of matched amino-acids
  - ANIKM matches AGIKM with 80% identity
- As we know, not all matches are equivalent − scoring matrices
- Insertions and deletions



query

database

www.ncbi.nlm.nih.gov/BLAST/

BLAST Results

Sequences producing significant alignments	Max score	Total score	Query cover	E value	Ident	Accession
beta-globin [Homo sapiens]	96.9	96.9	100%	5e-24	100%	AAP74754.1
mutant beta-globin [Homo sapiens]	96.9	96.9	100%	6e-24	100%	AAG46182.1
beta globin variant [Homo sapiens]	96.9	96.9	100%	6e-24	100%	AAP44006.1
hemoglobin beta [Homo sapiens]	96.9	96.9	100%	7e-24	100%	AFR11469.1
beta-globin thalassemia [Homo sapiens]	96.9	96.9	100%	1e-23	100%	AAA16335.1
beta-globin [Homo sapiens]	96.9	96.9	100%	1e-23	100%	AAA88069.1
truncated beta globin [Homo sapiens]	96.9	96.9	100%	1e-23	100%	ACF16769.1
beta globin [Homo sapiens]	96.9	96.9	100%	2e-23	100%	ACZ67952.1
beta globin [Homo sapiens]	96.9	96.9	100%	2e-23	100%	AAB60348.1
beta globin [Homo sapiens]	96.9	96.9	100%	8e-23	100%	AAC97372.1
hemoglobin beta chain [Homo sapiens]	96.9	96.9	100%	8e-23	100%	ADW79453.1
beta-globin [Homo sapiens]	96.9	96.9	100%	1e-22	100%	AAK30154.1
beta-globin [Homo sapiens]	96.9	96.9	100%	1e-22	100%	AAA88057.1
beta-globin [Homo sapiens]	96.9	96.9	100%	1e-22	100%	AAA99224.1

Heuristic Alignment Motivation

- $O(mn)$ too slow for large databases with high query traffic
- **Heuristic algorithm**: an algorithm that isn't guaranteed to find the optimal solution, but that is efficient and finds good solutions in practice
- heuristic methods do fast approximation to dynamic programming
  - FASTA [Pearson & Lipman, 1988]
  - BLAST [Altschul *et al*., 1990;
    Altschul et al., *Nucleic Acids Research* 1997]

Heuristic Alignment Motivation

- consider the task of searching SWISS-PROT against a query sequence:
  - say our query sequence is 362 amino-acids long
  - SWISS-PROT release 38 contained 29,085,265 amino acids
  - finding local alignments via dynamic programming would entail $O(10^{10})$ matrix operations
- many servers handle thousands of such queries a day (NCBI > 100,000)

How to query sequence database?

- Scan all sequences in database
- However
  - Results to queries need to be quick
  - Most sequences will be unrelated to query
  - Alignments need not be exact
- Exploit nature of the problem
  - If we are going to reject sequences with identity < 90%, then can quickly eliminate sequences if there aren't long stretches of amino acids in a row that match query
  - Pre-screen sequences for long stretches that match

BLAST Overview

- **B**asic **L**ocal **A**lignment **S**earch **T**ool
- BLAST heuristically finds high scoring local alignments
- typically used to search a query sequence against a database of sequences
- key tradeoff made: sensitivity vs. speed

$$\text{sensitivity} = \frac{\text{\# significant matches detected}}{\text{\# significant matches in DB}}$$

Overview of BLAST Algorithm

- given: query sequence *q*, word length *w*, word score threshold *T*, segment score threshold *S*
  - compile a list of "words" (of length *w*) that score at least *T* when compared to words from *q*
  - scan database for matches to words in list
  - extend all matches to seek high-scoring alignments
- return: alignments scoring at least *S*

Determining Query Words

Given:

query sequence: QLNFSAGW

word length $w = 2$ (default for protein usually $w = 3$)

word score threshold $T = 9$

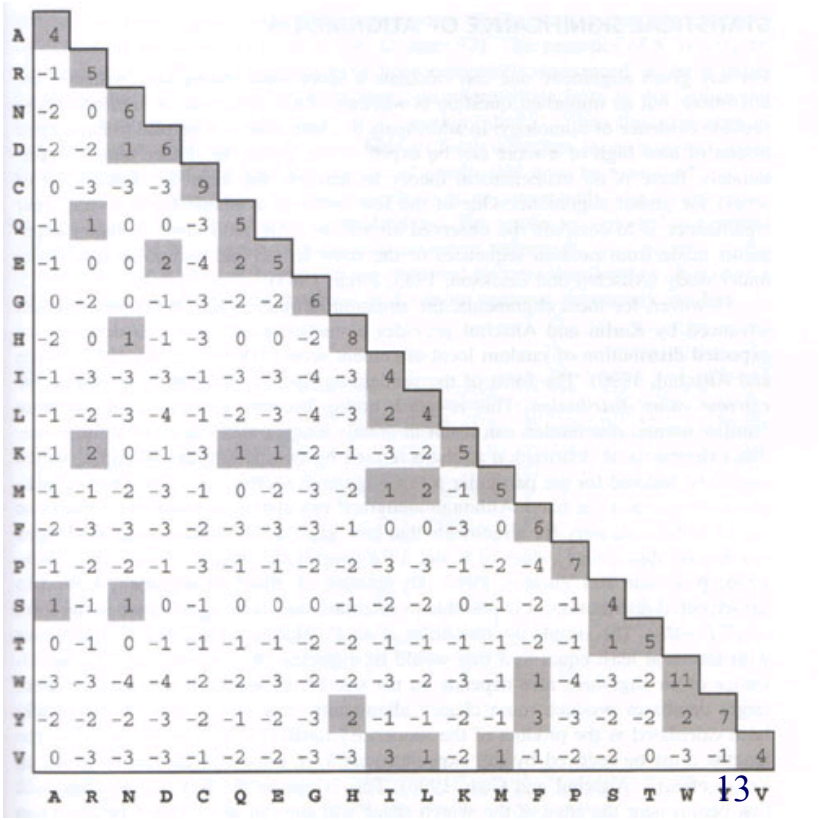
Step 1: determine all words of length w in query sequence (w-mers)

QL LN NF FS SA AG GW

Determining Query Words

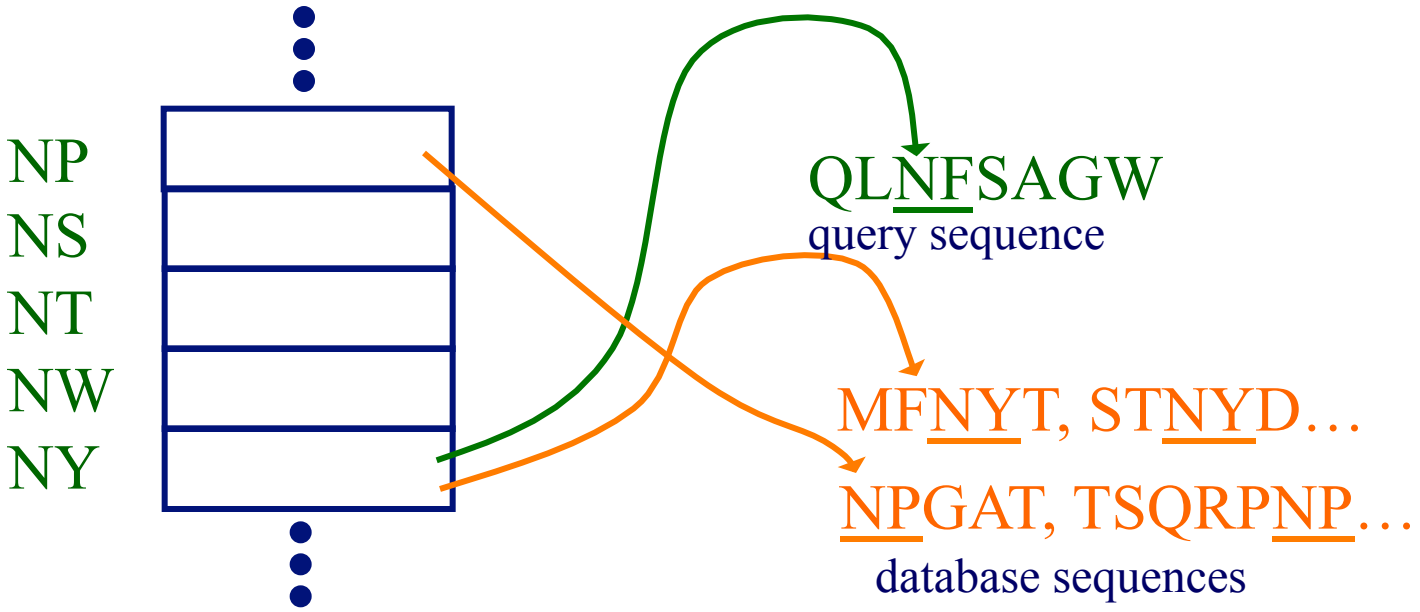
Step 2: determine all words that score at least *T* when compared to a word in the query sequence

words from sequence	query words w/ T=9
QL	QL=9
LN	LN=10
NF	NF=12, NY=9
…	
SA	none
...	

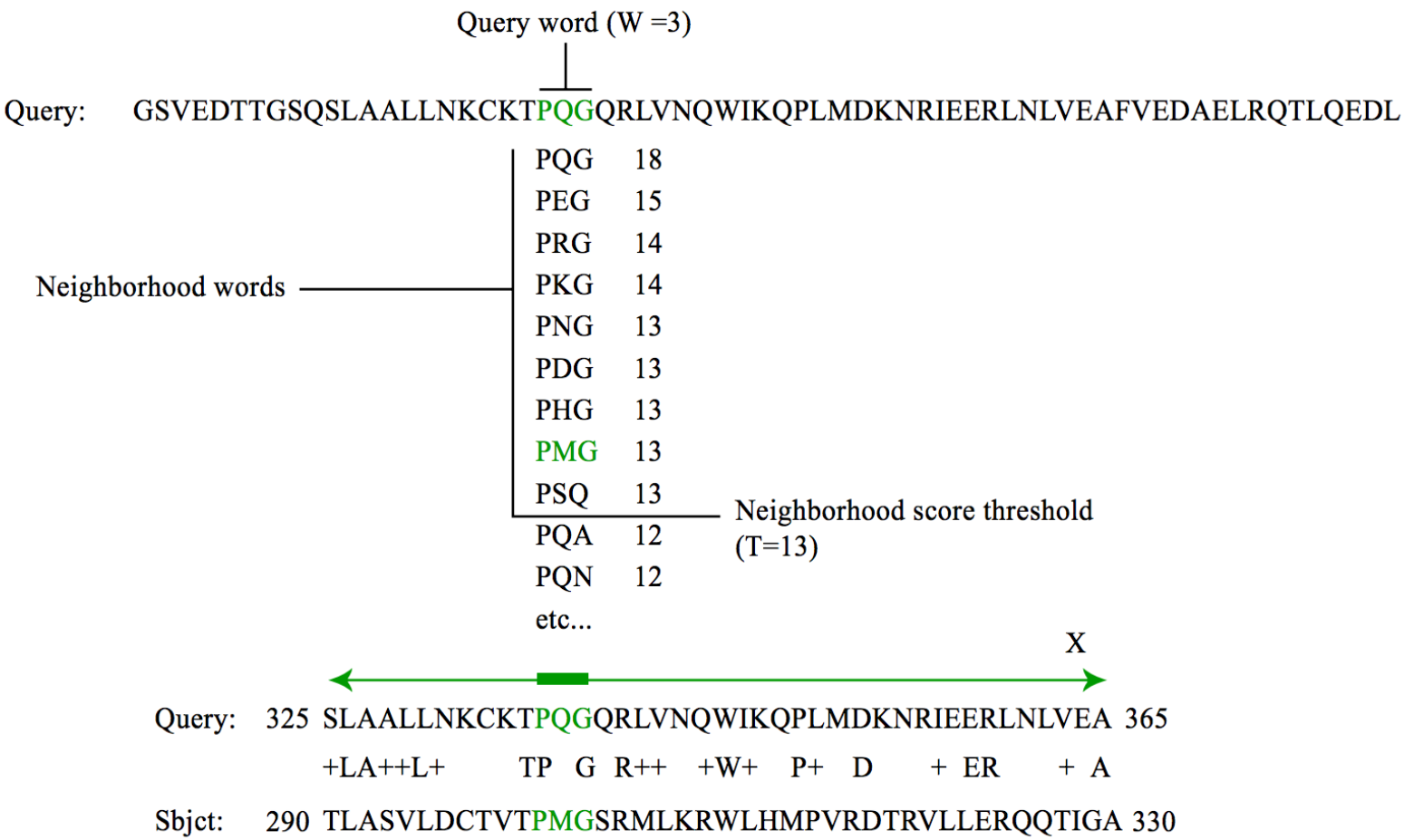


W-mer indexing

- Search database for all occurrences of query words
- Approach:
  - index database sequences into table of words (pre-compute this)
  - index query words into table (at query time)



THE BLAST SEARCH ALGORITHM

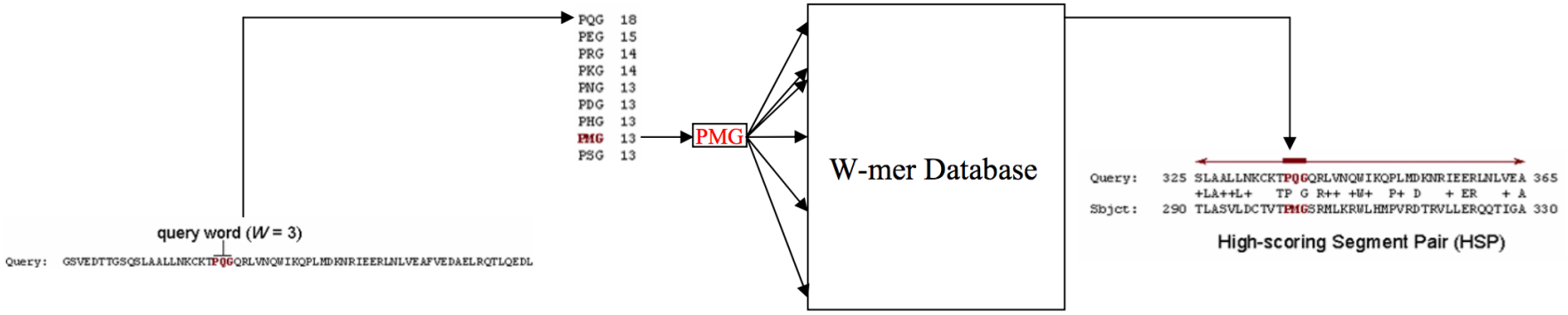


High-scoring Segment Pair (HSP)

Adapted from: Baxevanis, Andy. "Nucleotide and Protein Sequence Analysis I." Lecture presentation, National Human Genome Research

BLAST Algorithm

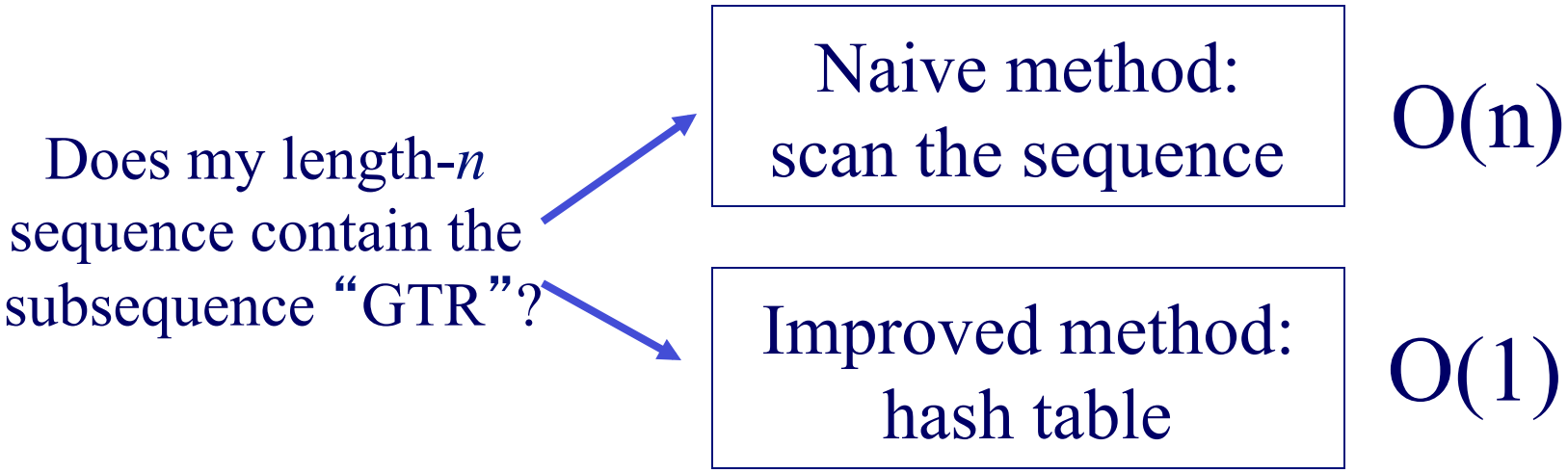
- Receive query
  - Split query into overlapping words of length W
  - Find neighborhood words for each word until threshold T
  - Look into table where these neighbor words occur: seeds



  - Extend seeds until score drops off under X
- Evaluate statistical significance of score
- Report scores and alignments

Data structure for quick lookup

- DP is O(nm); BLAST is O(m).
- employ a data structure to index the query sequence.
- data structure allows you to look up entries in a table in O(1) time.

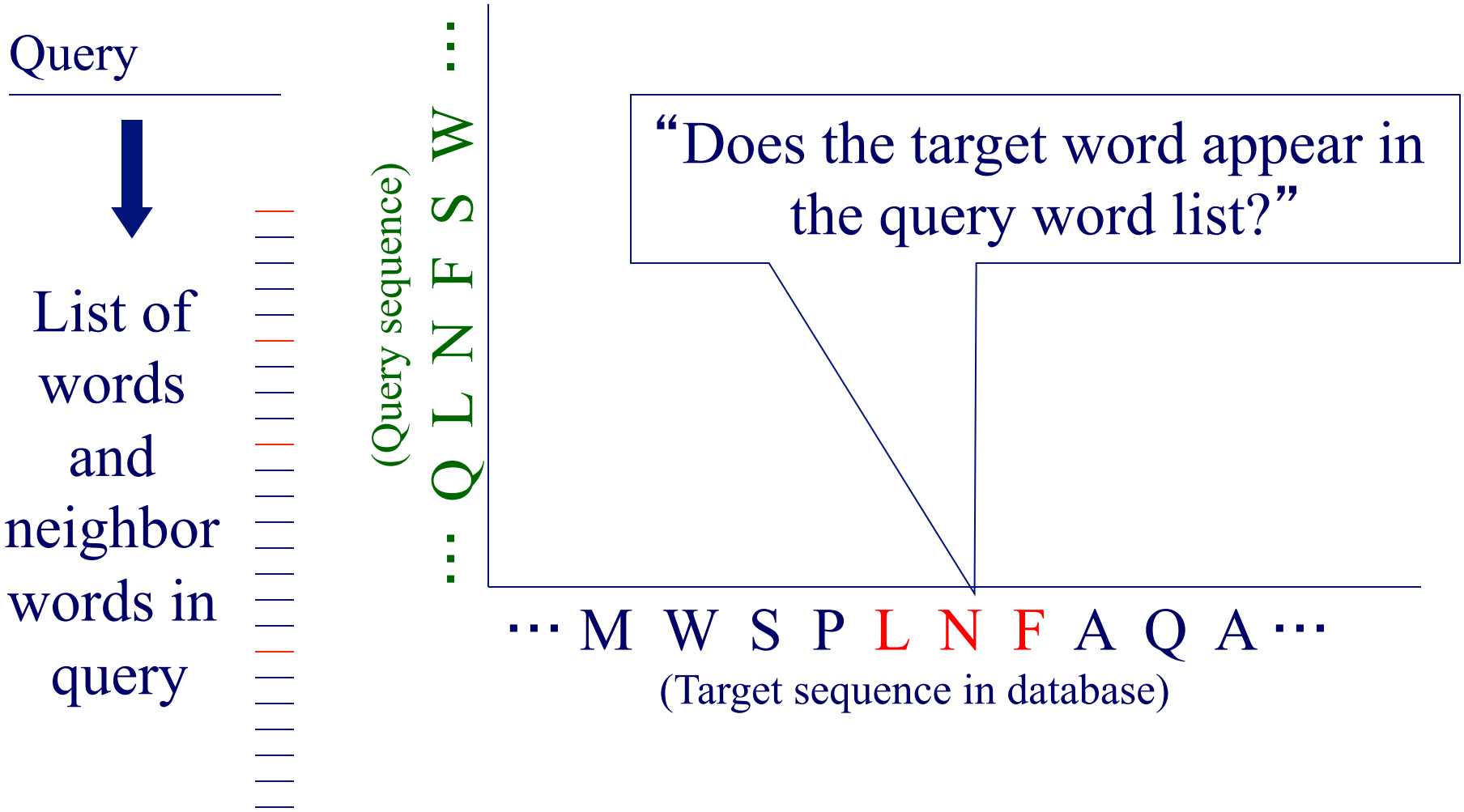


BLAST

Query

List of words and neighbor words in query

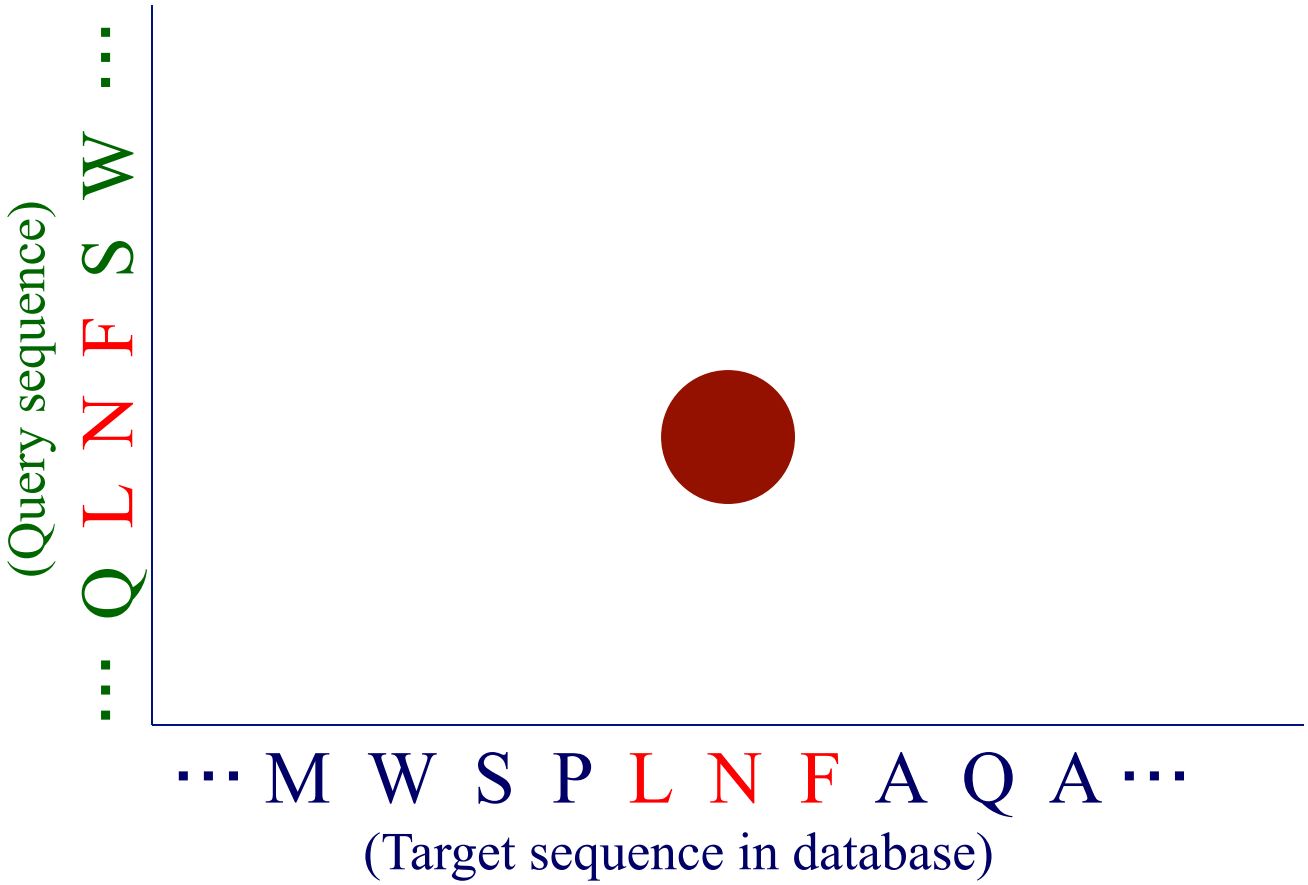
BLAST


Query
List of words and neighbor words in query
(Query sequence)
… Q L N F S W …
“Does the target word appear in the query word list?”
… M W S P L N F A Q A …
(Target sequence in database)

BLAST

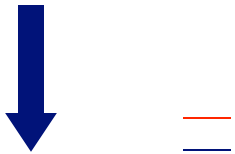
Query

List of words and neighbor words in query



BLAST

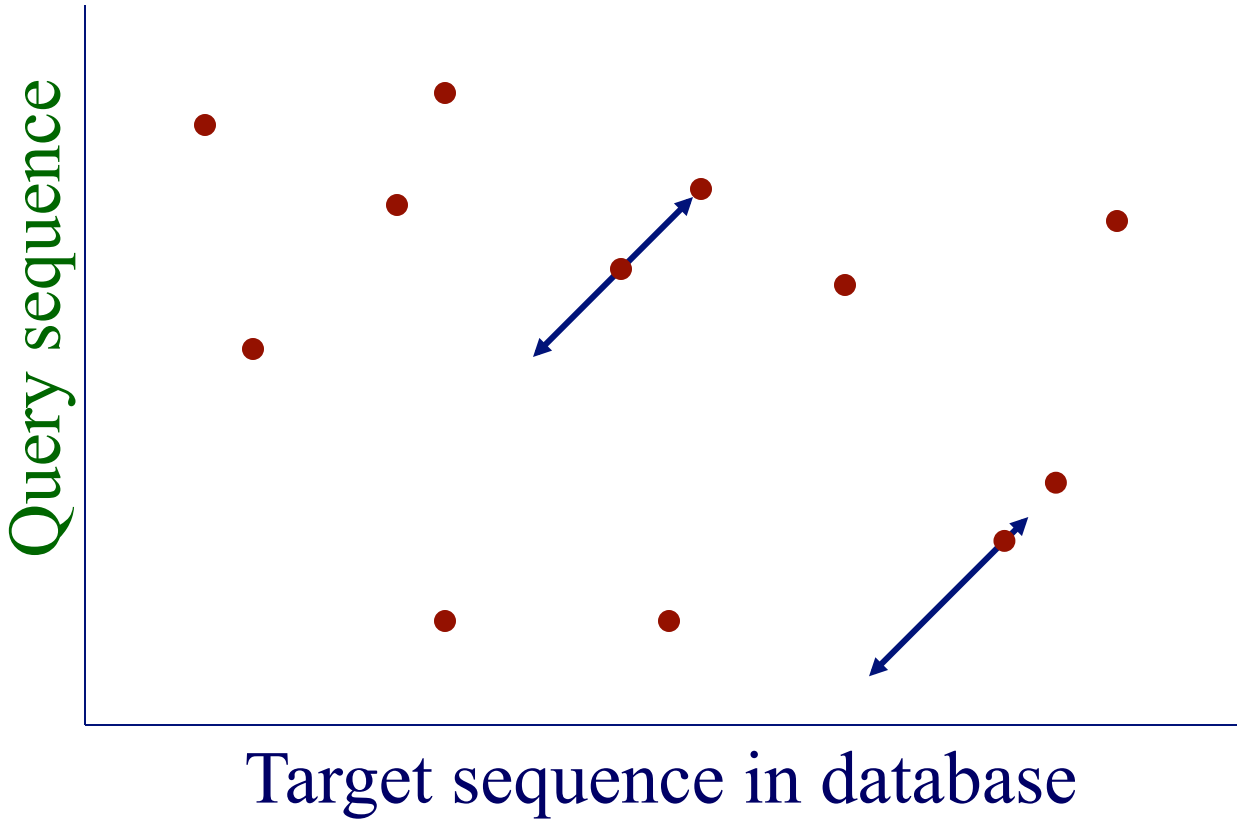
Query



List of words and neighbor words in query

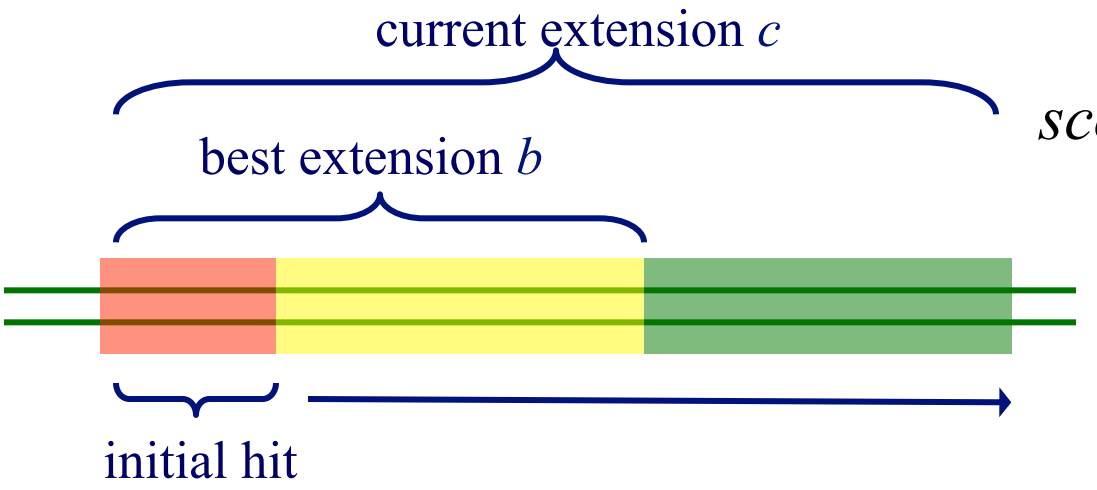
Extending hits

- BLAST extends hits into local alignments
- The original version of BLAST extended each hit separately



Extending Hits in Original Blast

- extend hits in both directions (without allowing gaps)
- terminate extension in one direction when cumulative score drops a certain distance below best score for shorter extensions



$score(c) \geq score(b) - \varepsilon$?

- return segment pairs scoring at least S

How to choose w and T?

- Tradeoff between running time and sensitivity
- Sensitivity

$$\text{sensitivity} = \frac{\#\text{ significant matches found}}{\#\text{ of significant matches in DB}}$$

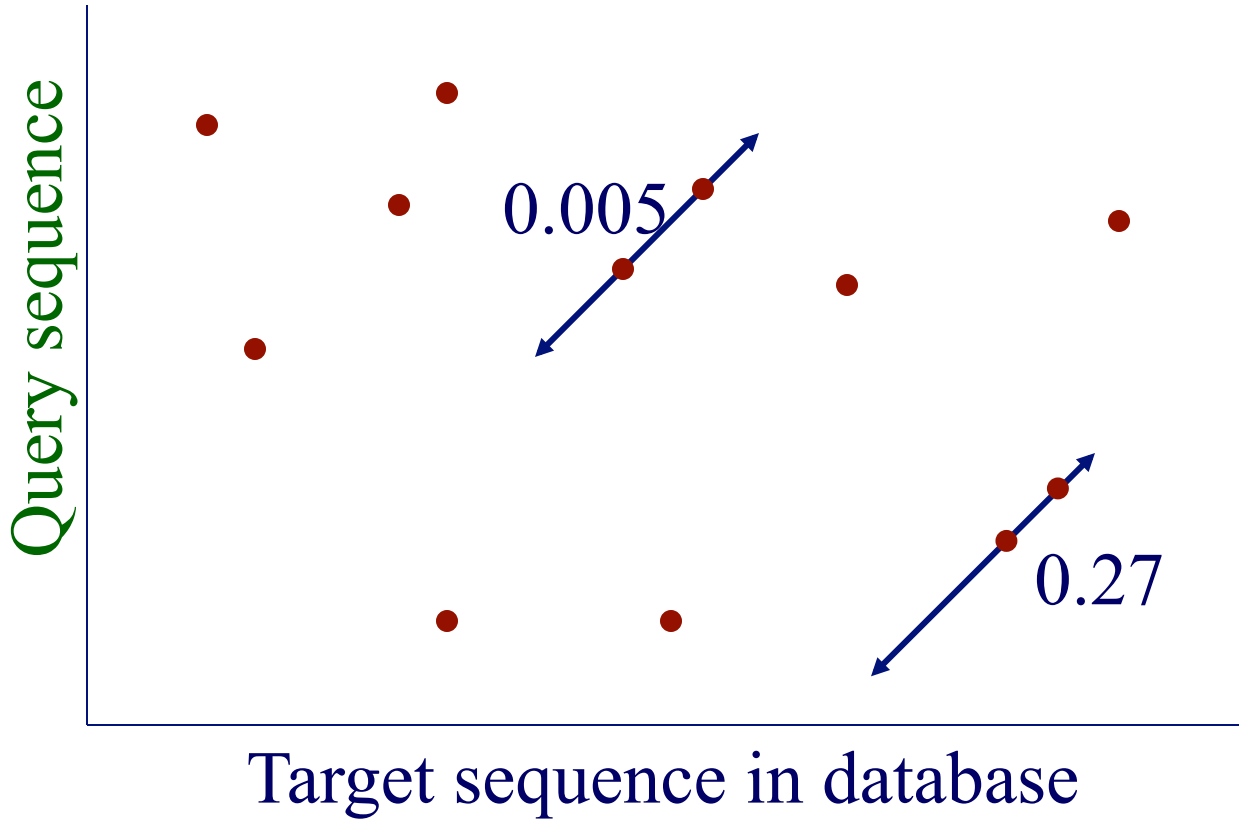
- T: most important parameter
  - small *T*: higher sensitivity, more hits to expand/compute
  - large *T*: lower sensitivity, fewer hits to expand
- *w*
  - Larger *w*: lower sensitivity, fewer hits to expand

The Two-Hit Method

- extension step typically accounts for 90% of BLAST's execution time
- key idea: do extension only when there are two hits on the same diagonal within distance A of each other
- to maintain sensitivity, lower *T* parameter
  - more single hits found
  - but only small fraction have associated 2nd hit

Extending hits

- Extend only if there are two hits on the same diagonal and within distance d of one another.



The Two-Hit Method

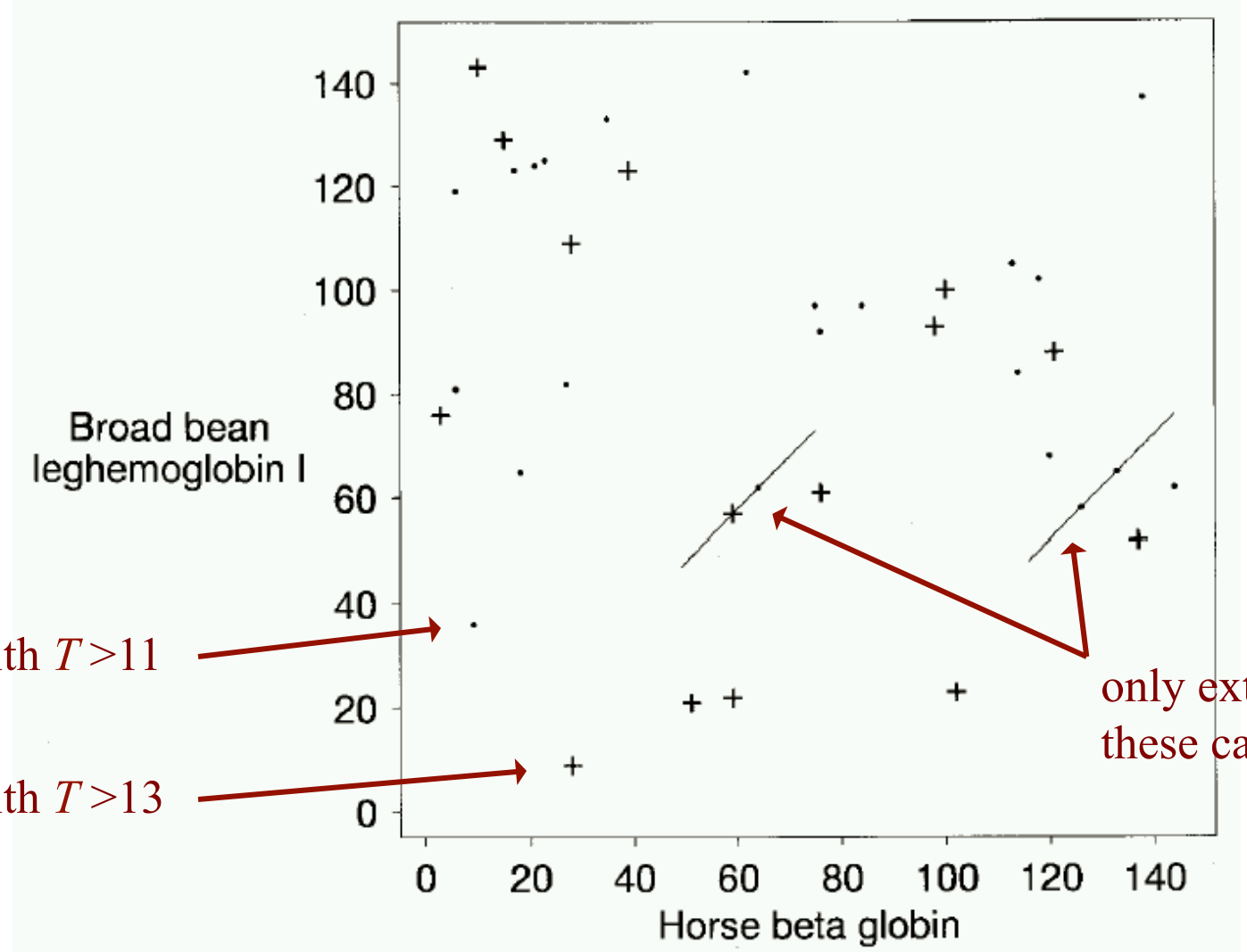


Figure from: Altschul et al. Nucleic Acids Research 25, 1997

Gapped BLAST

- trigger gapped alignment if two-hit extension has a sufficiently high score
- find segment with highest score; use central pair in this segment as seed
- run DP process both forward & backward from seed
- prune cells when local alignment score falls a certain distance below best score yet

Gapped BLAST

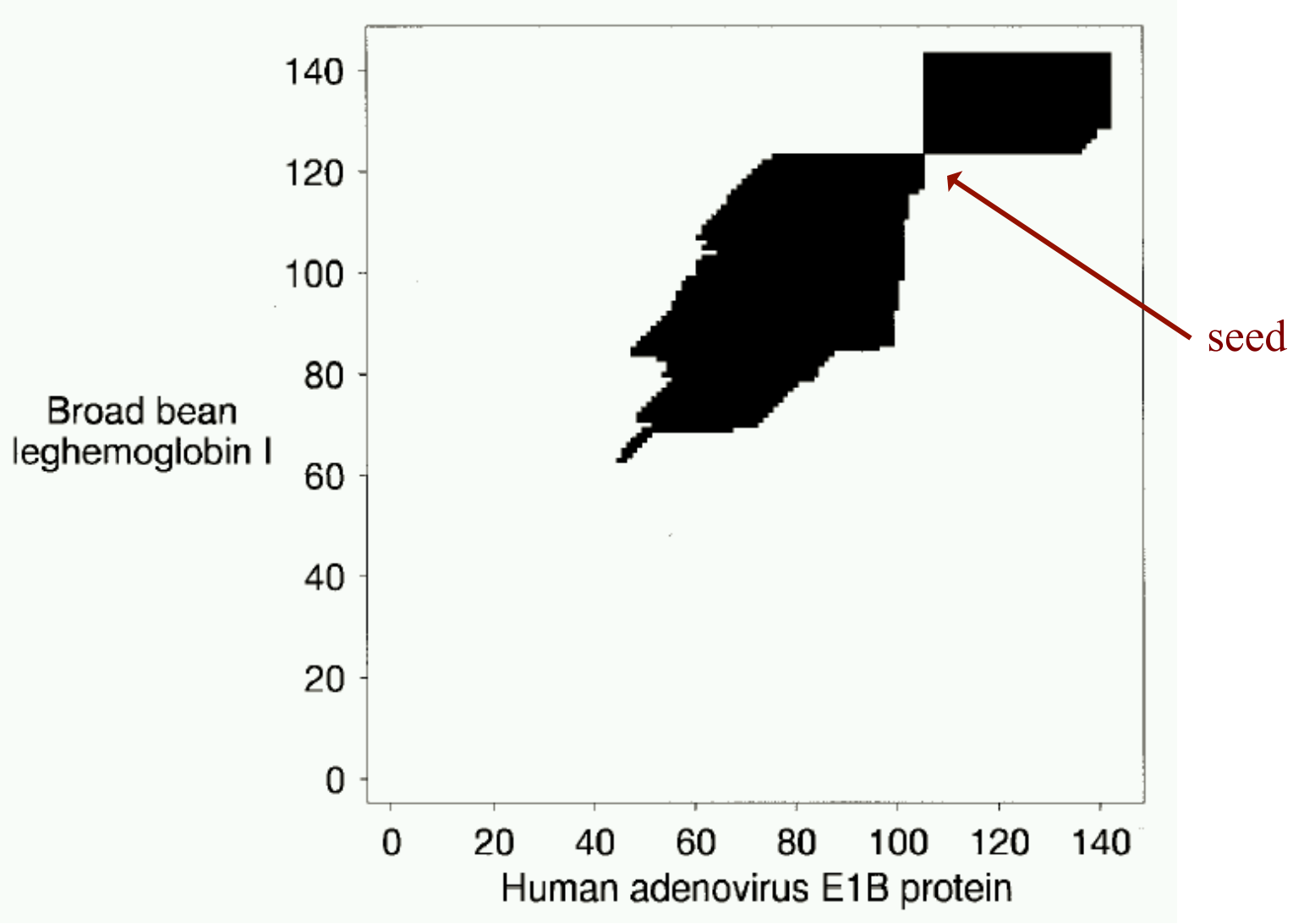


Figure from: Altschul et al. Nucleic Acids Research 25, 1997

When does BLAST fail?

```
ERDCRVSSFRVKENFDKARFAGTWYAMAKKDPEGLFLQDNIVAEFSVDENGHMSATAKGRVRLLNNWDVCADMVGTFTDT
E   R   F   E   K   A   Y   K   E   L   I   F   E   M   A   V   N   V   M   F
ECEIRQFLFIQRESARKEACATGTYREKKMDPELIVLVIWICPQFEQLEMRAMWIHAKJEVIUENAQCVIYTMQEPFCII
```

- BLAST works by joining together short regions of high similarity.
- Therefore, BLAST will fail to detect long regions of low similarity.

BLAST Programs

Program	Query	Database
BLASTP	Protein	Protein
BLASTN	DNA	DNA
BLASTX	Translated DNA	Protein
TBLASTN	Protein	Translated DNA
TBLASTX	Translated DNA	Translated DNA

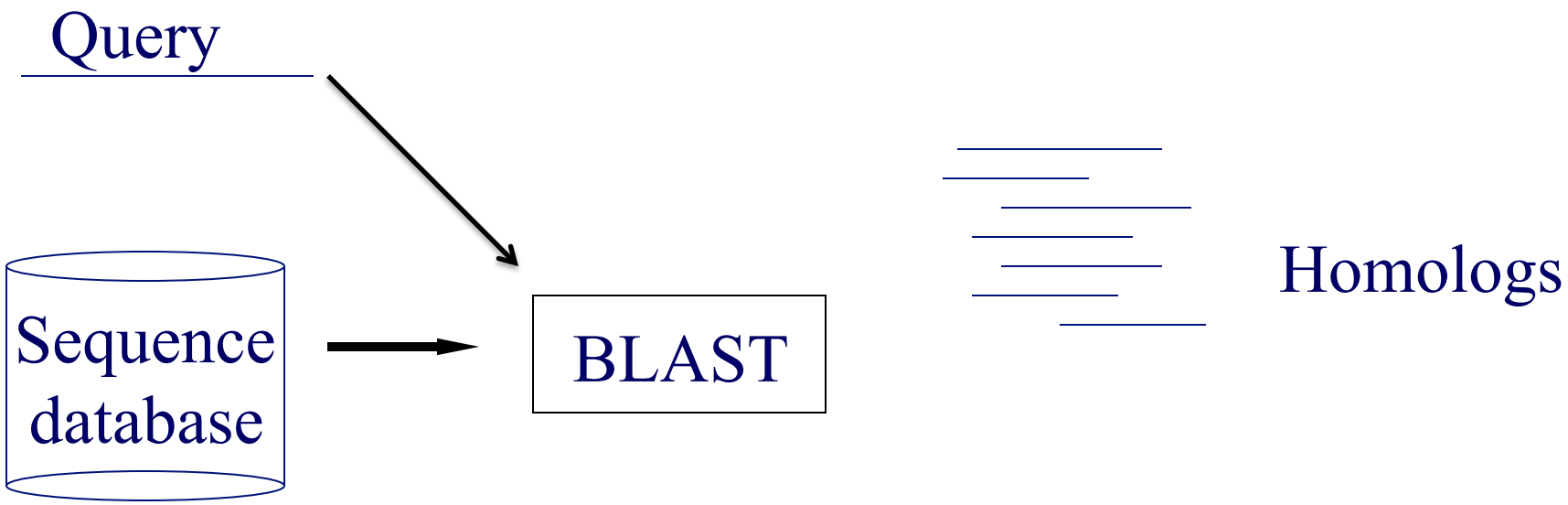
Summary of BLAST

- Dynamic programming is O(nm), where n is the length of the query and m is the size of the database. BLAST is O(m)
- BLAST produces an index of the query sequence that allows fast matching to the database
- Relative to Smith-Waterman, BLAST can produce *false negatives*; i.e., homologs that BLAST fails to detect, but is 10 to 50 times faster
- large impact:
  - NCBI's BLAST server handles more than 100,000 queries a day
  - most used bioinformatics program in the world

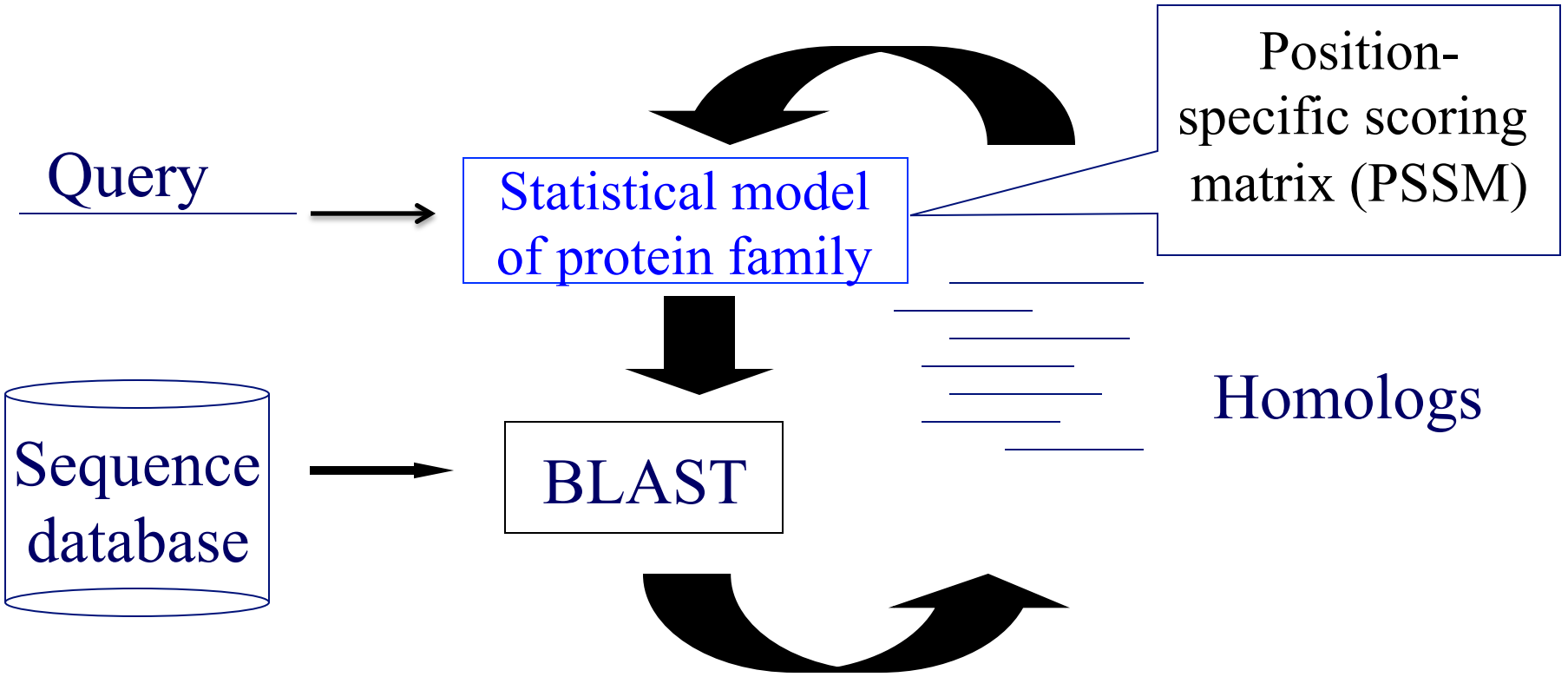
PSI (*Position Specific Iterated*) BLAST

- basic idea
  - use results from BLAST query to construct a *profile matrix*
  - search database with profile instead of query sequence
  - iterate

BLAST



Position-specific iterated BLAST



Position-specific scoring matrix

- A PSSM is an n by m matrix, where n is the size of the alphabet, and m is the length of the sequence.
- The entry at (i, j) is the score assigned by the PSSM to letter i at the jth position.

Position in query sequence

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

"K" at position 3 gets a score of 2.

Position-specific scoring matrix

- This PSSM assigns the sequence NMFWAFGH a score of 0 + -2 + -3 + -2 + -1 + 6 + 6 + 8 = 12.

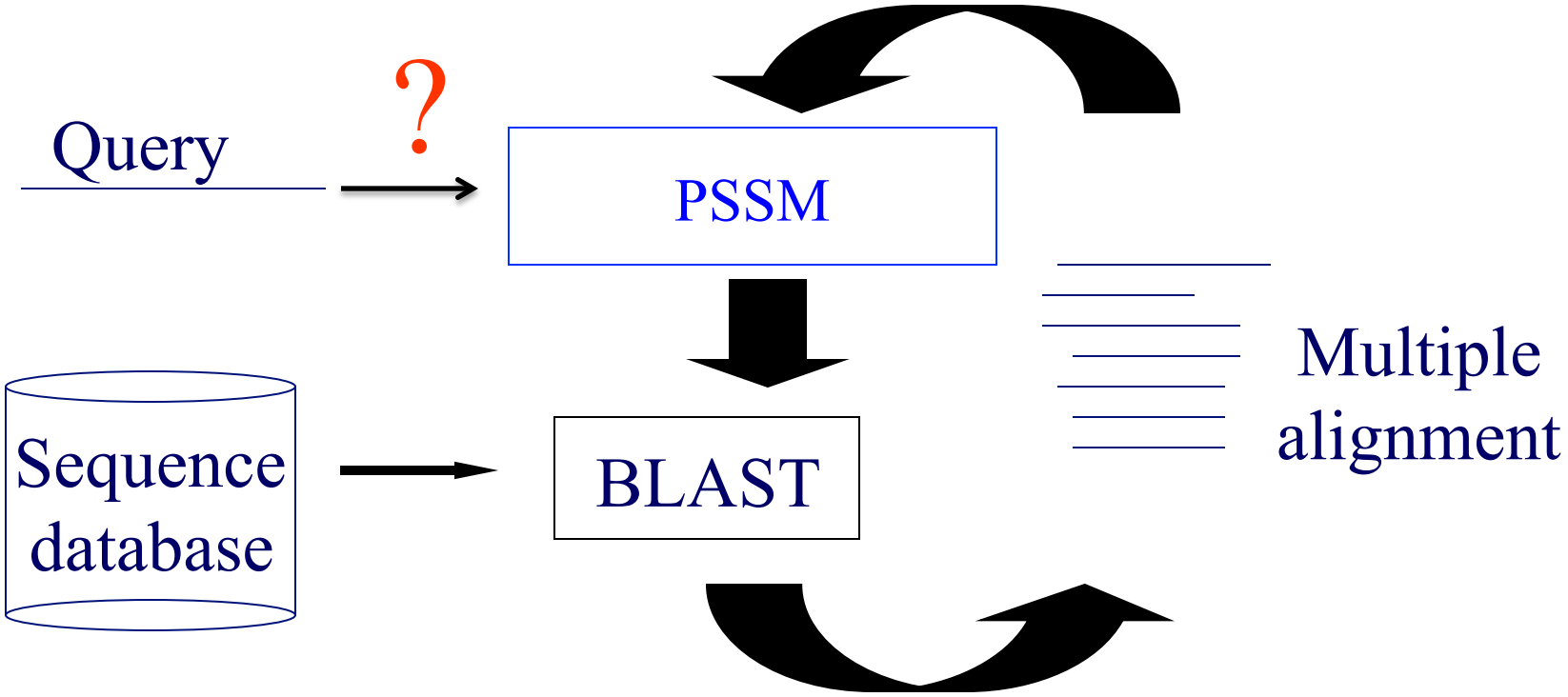
	N	M	F	W	A	F	G	H
A	-1	-2	-1	0	-1	-2	0	-2
R	5	0	5	-2	1	-3	-2	0
N	0	6	0	0	0	-3	0	1
D	-2	1	-2	-1	0	-3	-1	-1
C	-3	-3	-3	-3	-3	-2	-3	-3
Q	1	0	1	-2	5	-3	-2	0
E	0	0	0	-2	2	-3	-2	0
G	-2	0	-2	6	-2	-3	6	-2
H	0	1	0	-2	0	-1	-2	8
I	-3	-3	-3	-4	-3	0	-4	-3
L	-2	-3	-2	-4	-2	0	-4	-3
K	2	0	2	-2	1	-3	-2	-1
M	-1	-2	-1	-3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

- What score does this PSSM assign to KRPGHFLA?
- 2 + 0 + -2 + 6 + 0 + 6 + -4 + -2 = 6

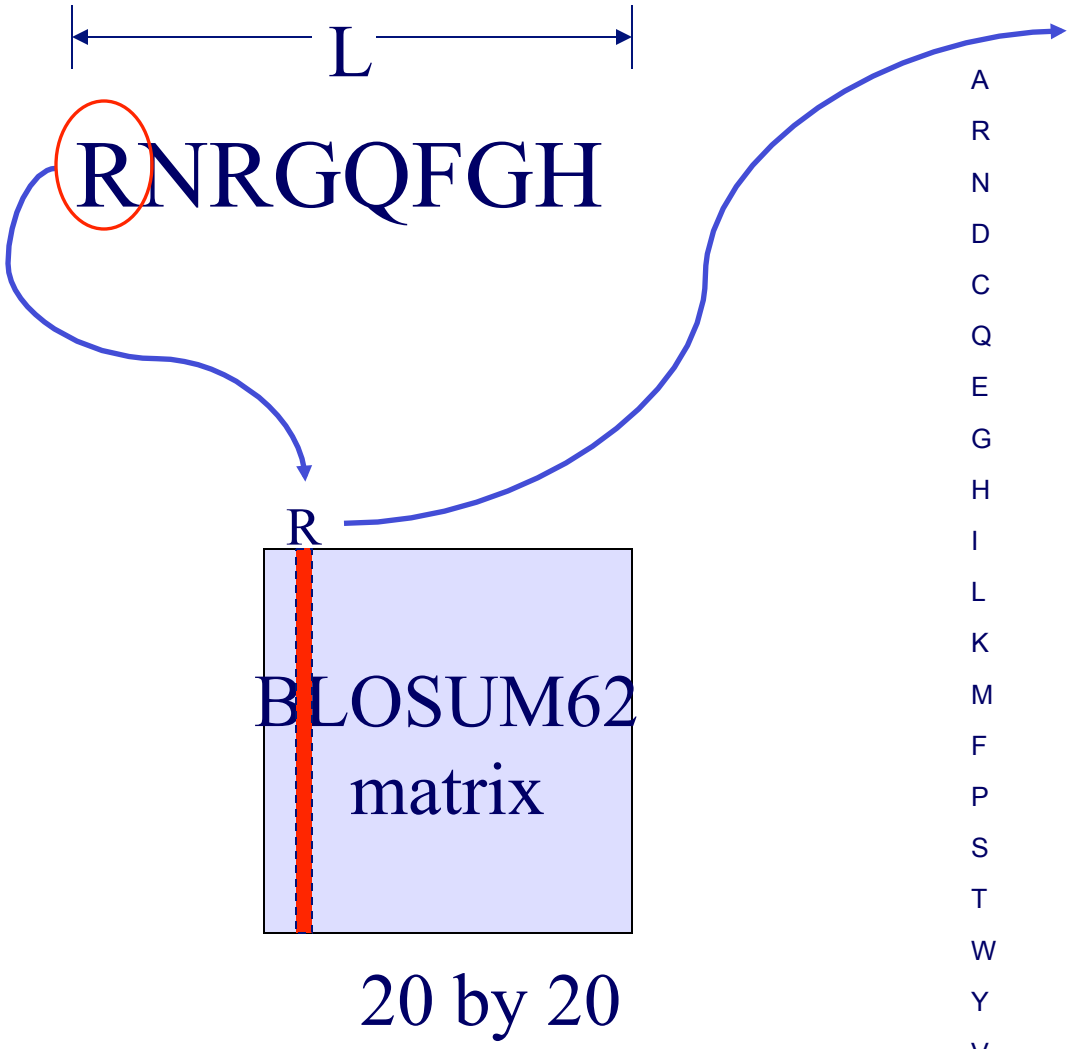
A	-1	-2	-1	0	-1	-2	0	-2
R	5	0	5	-2	1	-3	-2	0
N	0	6	0	0	0	-3	0	1
D	-2	1	-2	-1	0	-3	-1	-1
C	-3	-3	-3	-3	-3	-2	-3	-3
Q	1	0	1	-2	5	-3	-2	0
E	0	0	0	-2	2	-3	-2	0
G	-2	0	-2	6	-2	-3	6	-2
H	0	1	0	-2	0	-1	-2	8
I	-3	-3	-3	-4	-3	0	-4	-3
L	-2	-3	-2	-4	-2	0	-4	-3
K	2	0	2	-2	1	-3	-2	-1
M	-1	-2	-1	-3	0	0	-3	-2
F	-3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

How PSI-BLAST makes PSSMs

Position-specific iterated BLAST



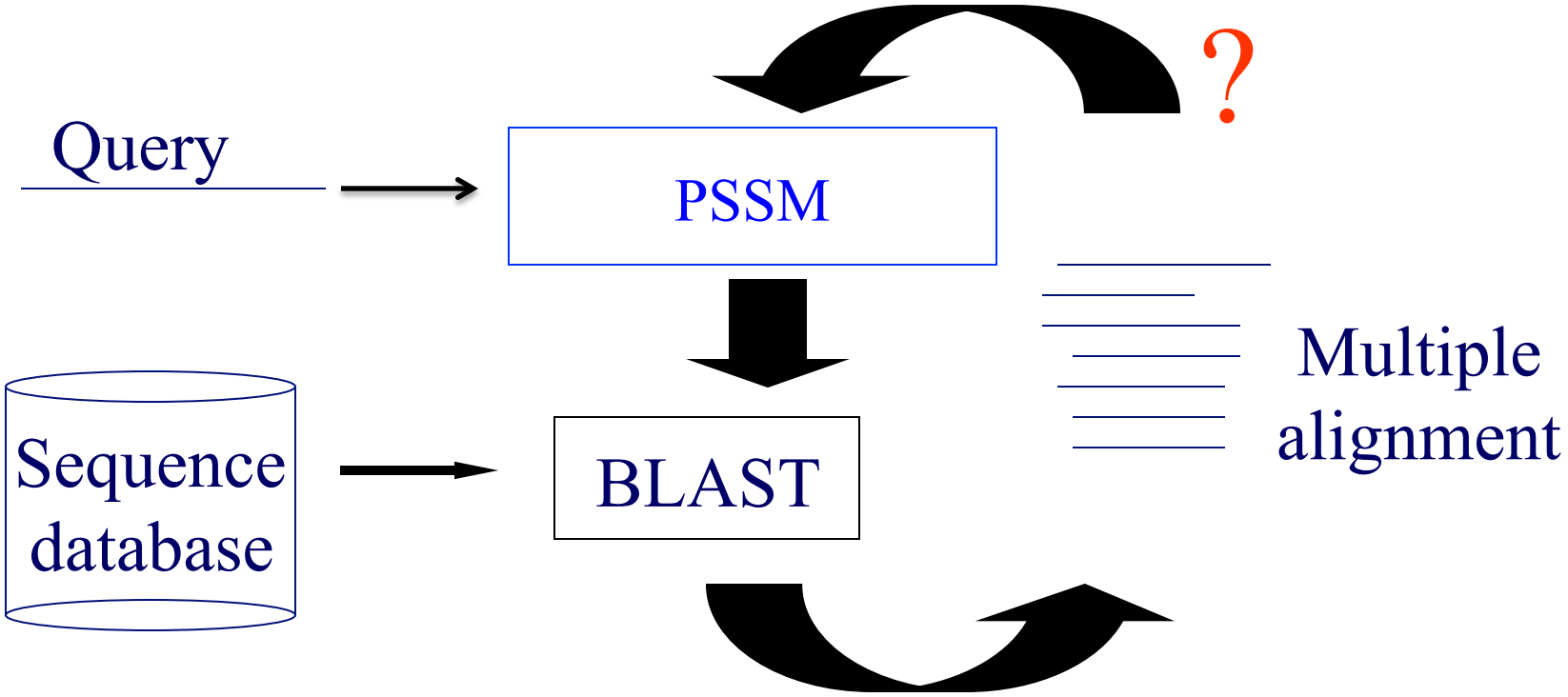
Creating a PSSM from 1 sequence



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

20 by L

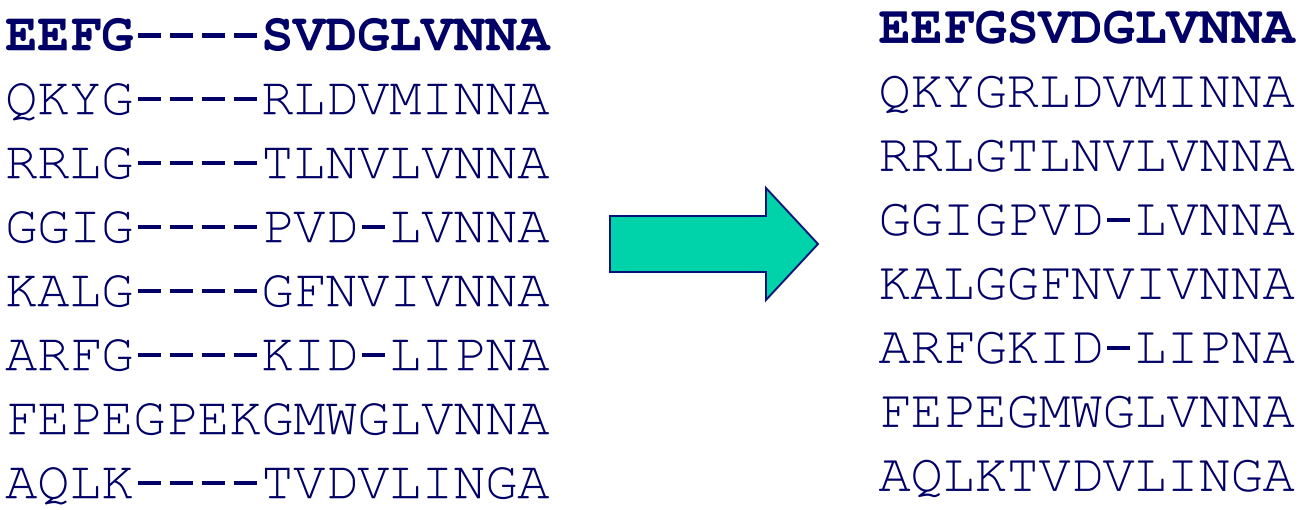
Position-specific iterated BLAST



Creating a PSSM from multiple sequences

- Discard columns that contain gaps in the query.
- For each column C
  - Compute relative sequence weights
  - Compute PSSM entries, taking into account
    - Observed residues in this column
    - Sequence weights
    - Substitution matrix

Discard query gap columns

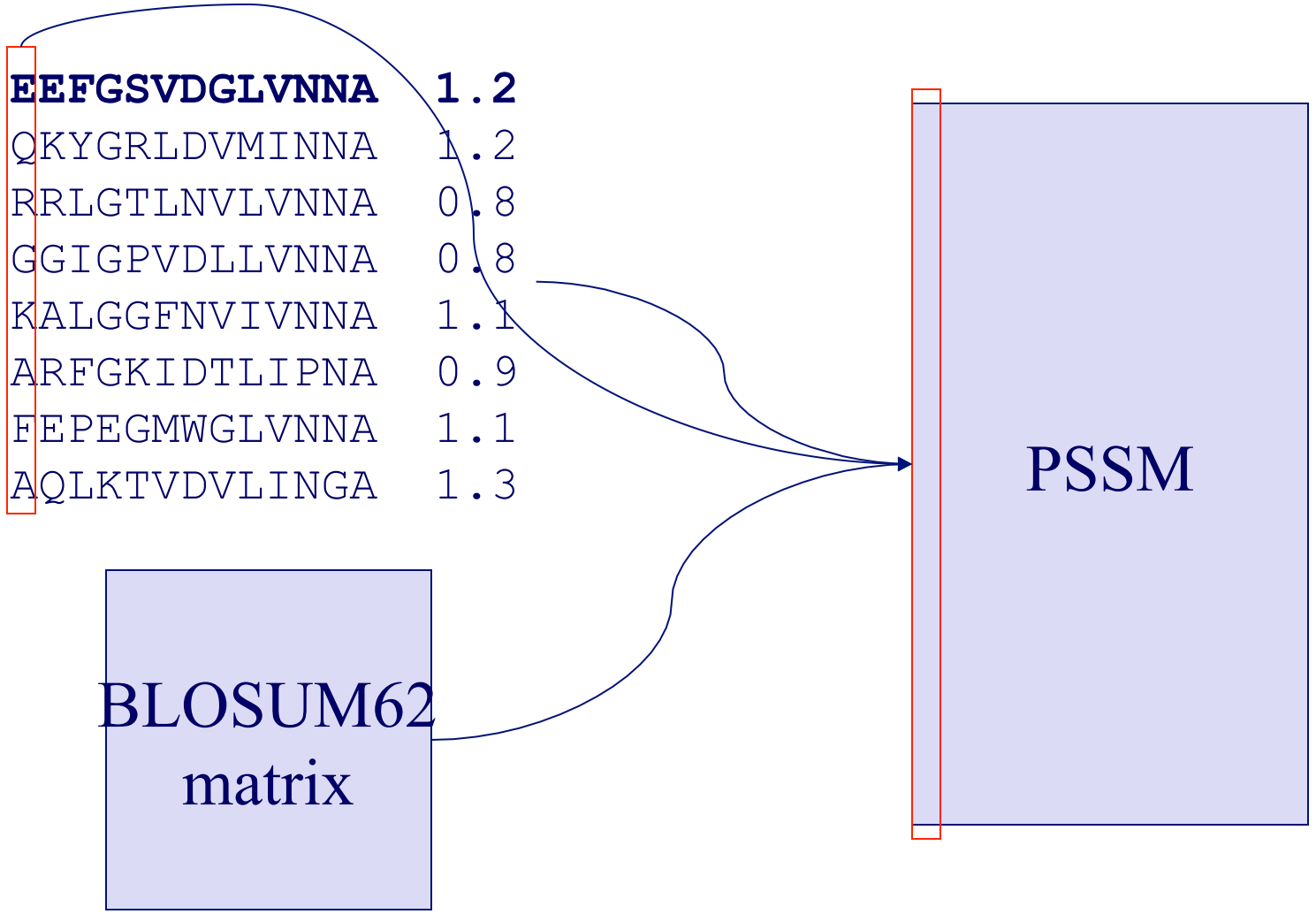


Compute sequence weights

EEFGSVDGLVNNA	**1.2**
QKYGRLDVMINNA	1.2
RRLGTLNVLVNNA	0.8
GGIGPVDLLVNNA	0.8
KALGGFNVIVNNA	1.1
ARFGKIDTLIPNA	0.9
FEPEGMWGLVNNA	1.1
AQLKTVDVLINGA	1.3

- Low weights are assigned to redundant sequences.
- High weights are assigned to unique sequences.

Compute PSSM entries



PSI BLAST: Constructing the Profile Matrix

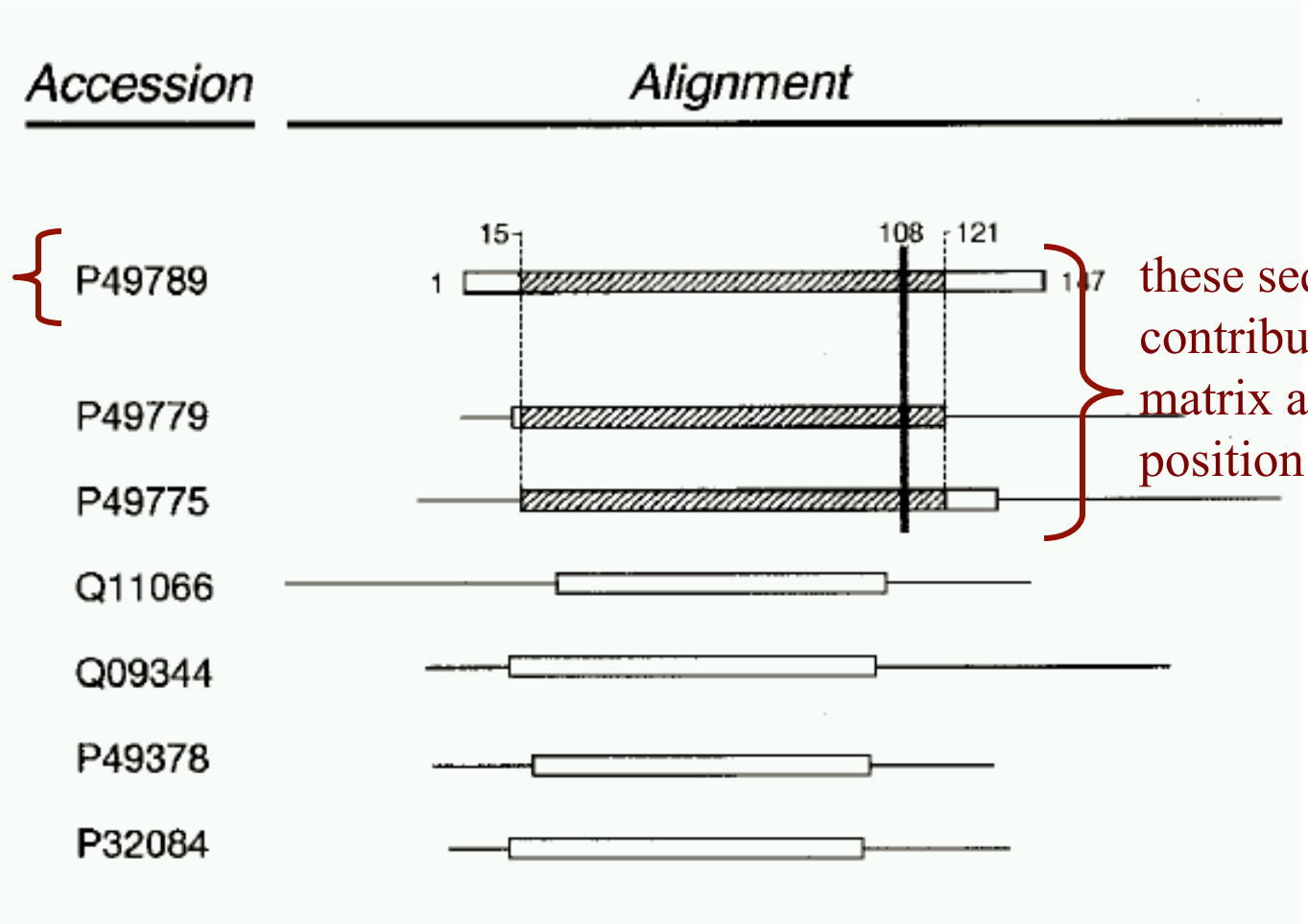
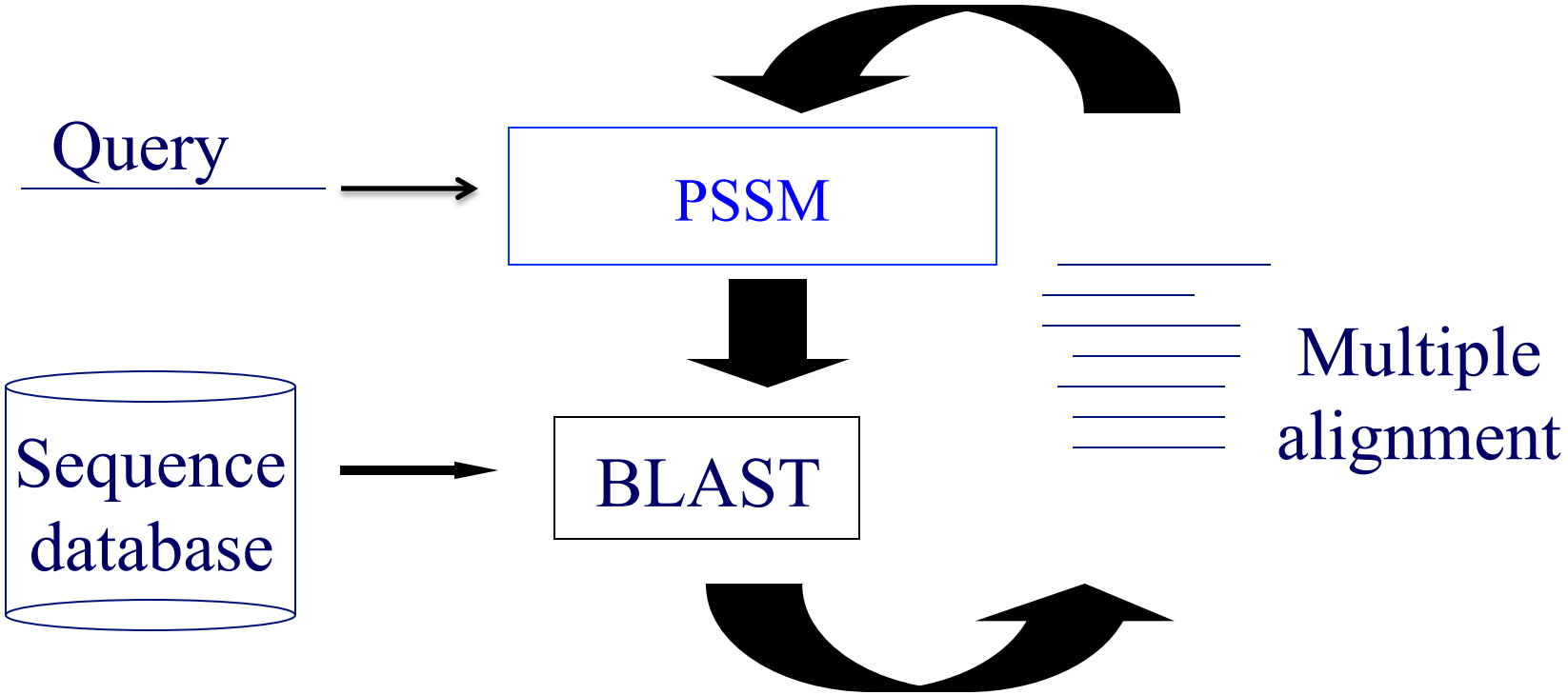
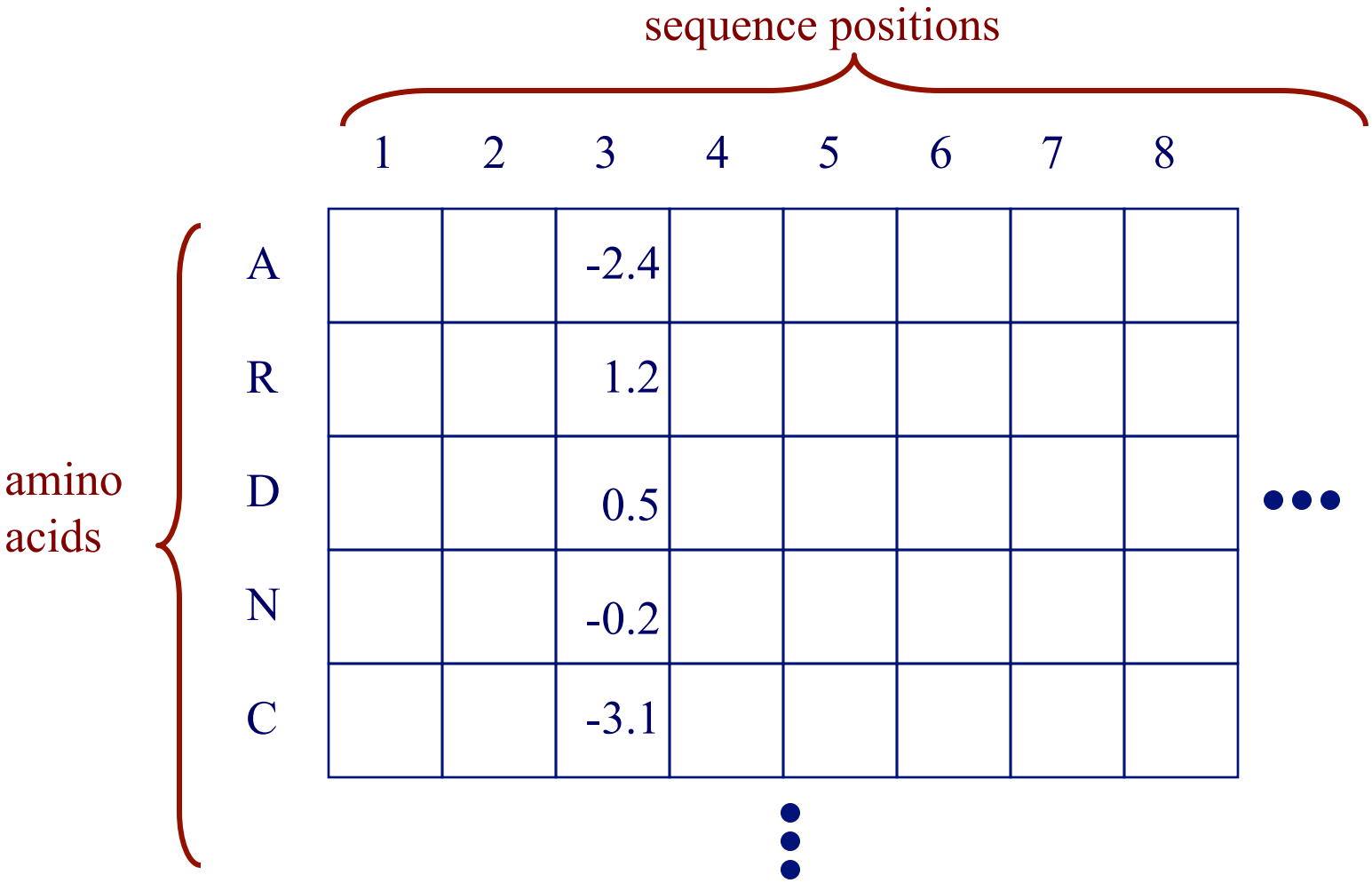


Figure from: Altschul et al. Nucleic Acids Research 25, 1997

Position-specific iterated BLAST

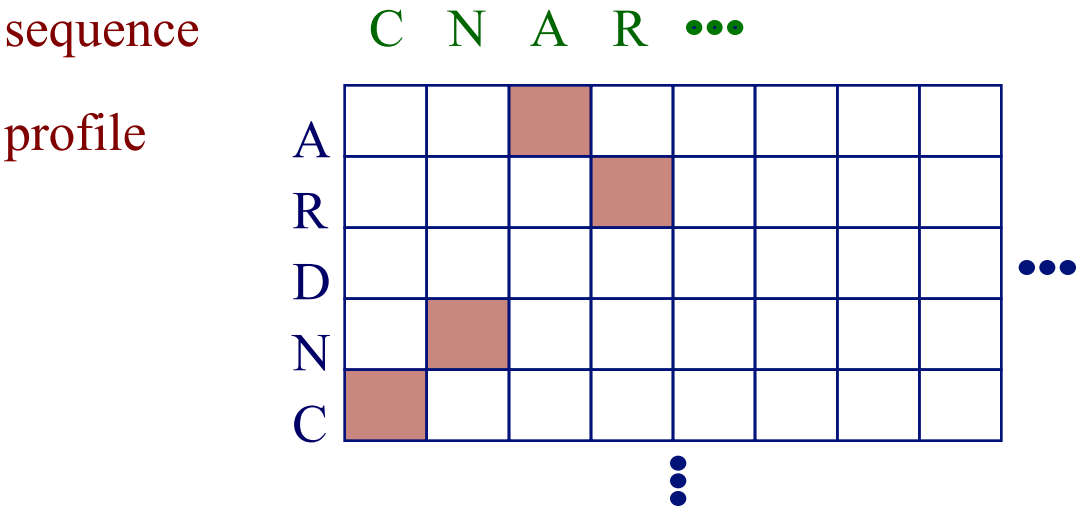


A Profile Matrix



PSI BLAST: Searching with a Profile

- aligning profile matrix to a simple sequence
  - like aligning two sequences
  - except score for aligning a character with a matrix position is given by the matrix itself – not a substitution matrix



Summary of PSI-BLAST

- PSI-BLAST builds a model of the query sequence and its close homologs.
- Instead of comparing a target sequence to the query, each target is compared to the model.
- The PSI-BLAST model is called a position-specific scoring matrix (PSSM).
- The PSSM can be constructed from a collection of targets aligned to the query sequence.
- PSI-BLAST is more accurate than BLAST.