Bioinformatics Algorithms
COS-BIOL-530/630
Lecture02

Days & Times	Room	Meeting Dates
Tu 2:00PM - 3:50PM	Thomas Gosnell Hall (GOS)-2178	01/13/2025 - 04/28/2025
Th 2:00PM - 3:50PM	Thomas Gosnell Hall (GOS)-2178	01/13/2025 - 04/28/2025

Instructor:
Fernando Rodriguez
email: frvsbi@rit.edu
Office: Orange Hall 1311

Bioinformatics Algorithms
COS-BIOL-530/630
Lecture02

Announcements
Week2:
Lecture02
Lab02
- Discussion 2
- Activity 2

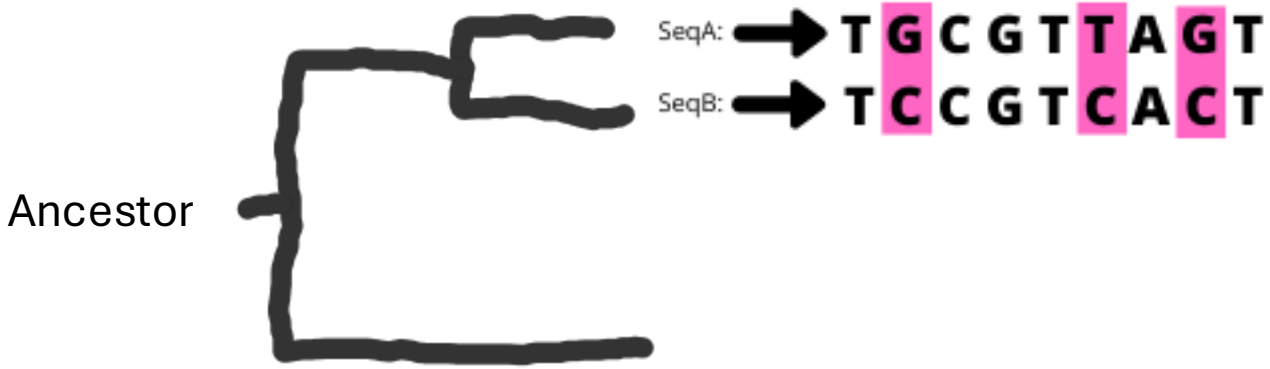
Activity1 due on Thursday

Quiz 2 opens on Friday. Due on Tuesday 2pm

- Let's talk about **Perusal**
- RIT Show Case (April 10th): **Exam2**
  - **Move it to April 15th?**
- Env. Genomics?

The biological problem: pairwise sequence alignment

- Given two DNA or protein sequences, we want to infer if they are homologous (or not).



Sequence alignments

Classes of alignment

- Pairwise vs multiple alignment
- Local vs. global alignment
- Exhaustive vs approximate

Historic context

Global, exhaustive, pairwise alignment
– Needleman S.B. and Wunsch C.D. (1970) J. Mol. Biol. 48, 443-453
Local, exhaustive, pairwise alignment
– Smith T.F. and Waterman M.S. (1981) J. Mol. Biol.147, 195-197
Local, approximate, pairwise alignment*
– BLAST: Basic Local Alignment Search Tool (Altschul et al. 1990).
Global, approximate, multiple alignment
– ClustalW: Thompson,J.D. et al. (1994) Nucleic Acids Res., 22, 4673–4680.
Local, approximate, pairwise alignment*
– Bowtie/BWA: time/memory solution to short read mapping (2009/2009)

*
Start with a string search "a seed"

Pairwise sequence alignment is a fundamental operation of bioinformatics

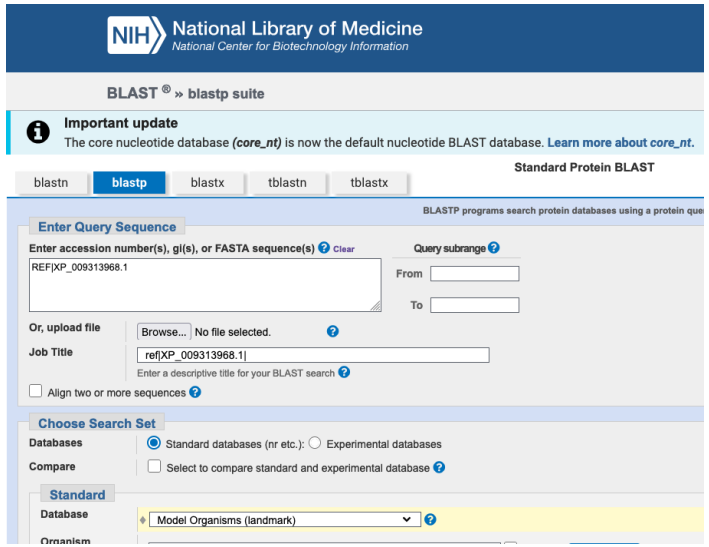
- It is used to decide if two proteins (or genes) are related structurally or functionally.
- It is used to identify domains or motifs that are shared between proteins.
- It is the basis of BLAST search (Lab01).
- It is used every time in the analysis of genomes.

Alignments algorithms
- Lecture02 & Lecture03 -

- Pairwise Alignment
  - Global
  - Local
  - Scoring system
  - Dynamic programming

- Multiple Sequence Alignment (MSA)

- Heuristic – Database search
  - BLAST



Pairwise alignment definition

What is the difference between similarity, identity, and homology?

a) They describe the same concept.

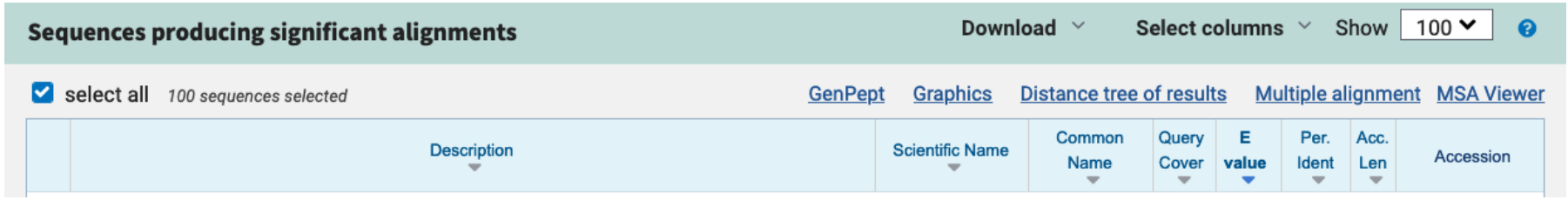
b) Homology is the percentage of similarity between two sequences.

c) The three, similarity, identity, and homology, can be quantified (number).

Pairwise alignment definition

The process of lining up two sequences to achieve maximal levels of **identity** (and conservation, in the case of amino acid sequences) to assess the degree of **similarity** and the possibility of **homology**.

...we need to set a few concepts first.



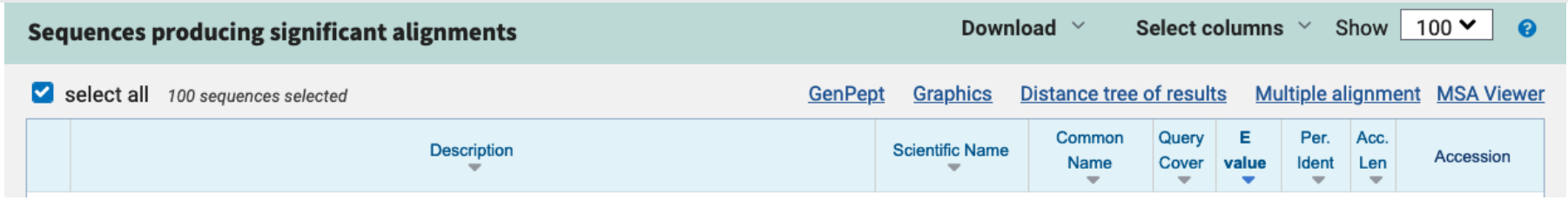
Pairwise alignment definition

Homology:
Sequence homology refers to the degree of similarity in the genetic sequences of different proteins or molecules. Homology indicates an ancient common origin.

Identity:
It is the number of characters that match precisely between two different sequences (Gaps are not counted). 100% identity does not mean two sequences are the same.

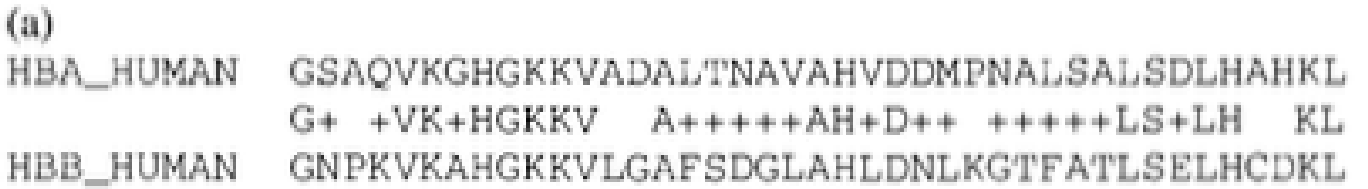
Seq1 AAGGCGTT
Seq2 AAGGCG

Similarity:
The extent to which nucleotide or protein sequences are related. The optimal matching algorithm finds the minimal number of edit operations.



Pairwise alignment

- How to determine two sequences are related?
  - First, align the sequences (or part of them)
  - Second, determine if the are related



Biologically meaningful alignment

Human alpha globin vs. beta globin protein sequence
(+) similar positions (score matrix)

Pairwise alignment

- How to determine two sequences are related?
  - First, align the sequences (or part of them)
  - Second, determine if the are related

(a)

```
HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
            G+ +VK+HGKKV  A+++++AH+D++ +++++LS+LH  KL
HBB_HUMAN   GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL
```

✓ Biologically meaningful alignment

(b)

```
HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
            ++ ++++H+ KV   + +A  ++           +L+ L+++H+ K
LGB2_LUPLU NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG
```

✓ Biologically meaningful alignment

(c)

```
HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSD----LHAHKL
            GS+ + G +   +D L  ++ H+ D+  A +AL D    ++AH+
F11G11.2    GSGYLVGDSLTFVDLL--VAQHTADLLAANAALLDEFPQFKAHQE
```

✗ No Biologically meaningful alignment

Edited from Biological sequence analysis, Durbin.

Pairwise alignment

- How to determine two sequences are related?
  - First, align the sequences (or part of them)
  - Second, determine if the are related

- Related? This is the challenge for pairwise alignment methods

- Definitions:

  - Global vs. Local sequence alignments
  - Scoring system

Global vs. Local sequence alignments

Methods used to assess sequence similarity (and to infer homology) can be grouped into two major types.

- Global sequence alignment
  - Compare two sequences along their entire length
  - **Find the best alignment**
  - Mainly applicable to highly similar sequences with equal length
  - Low similarity sequences will miss important biological information

- Local sequence alignment
  - Most biology approaches depend on local alignments
  - Find the **most similar regions** between two seqs (not trying to align the entire length)

Global vs. Local sequence alignments

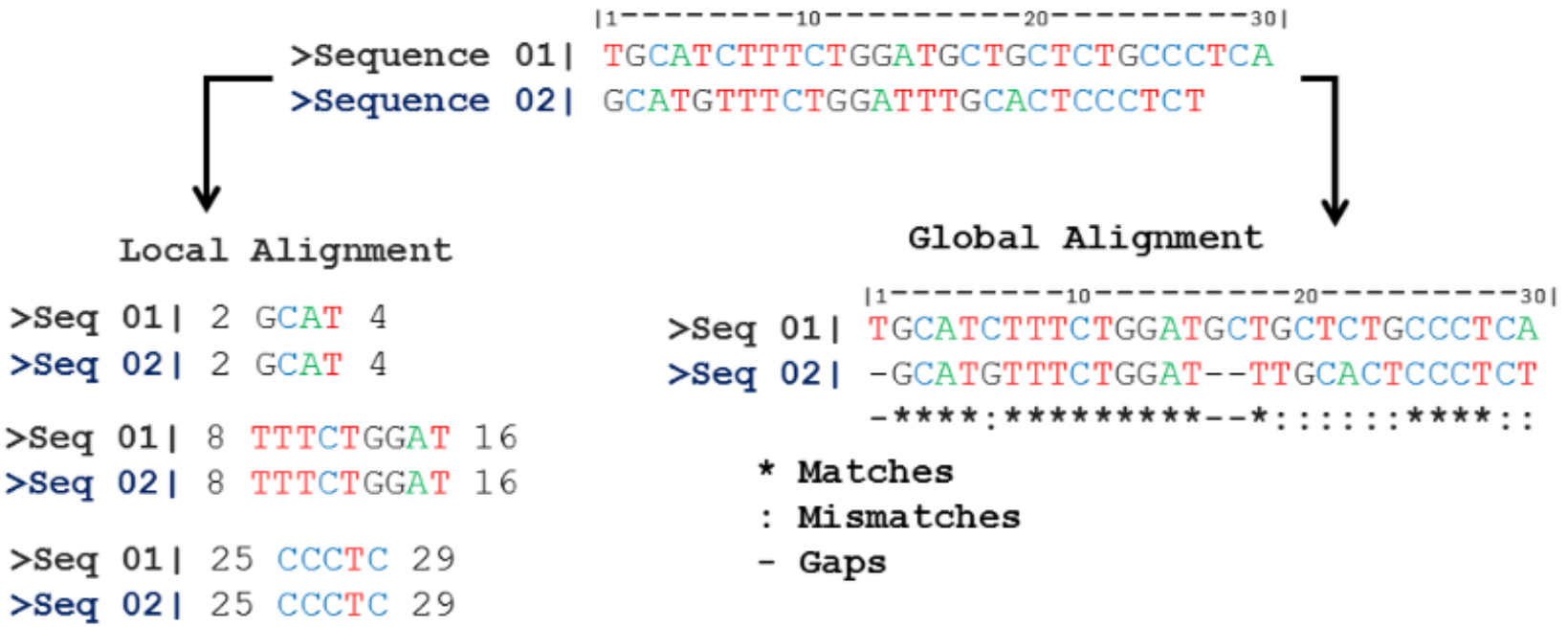


Figure modified from Prosdocimi *et al*. (2002)

Scoring system

- To infer homology, we will **calculate a score** that reflects how similar the two sequences are.
- The basic mutation (molecular) process has:
  - No mutation/Match
    - Positive value in the score
  - Substitutions
    - Non-positive value (or null value) for mismatch
  - Insertions/Deletions which add or remove residues (gaps)
    - Penalize gaps (negative value)

(b)

```
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
           ++ ++++H+ KV    + +A   ++           +L+ L+++H+ K
LGB2_LUPLU NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG
```

Scoring system

- The total score assigned to an **alignment** will be the sum of terms for each aligned pair of residues, plus terms of each gap.

- We use an **additive scoring scheme** in the algorithms to find optimal alignments.

- Much effort has been devoted to the development of constructs called **scoring matrices.**

- These matrices are empirical weighting schemes, which three major biological factors (proteins).
  - Conservation: between proteins and conservative substitutions (physicochemical standpoint).
    - Residue charge
    - Size
    - Hydrophobicity

*Link to amino acid physical properties

  - Frequency: mostly based on observation.
  - Evolution: By design, scoring matrices implicitly represent evolutionary patterns, and matrices can be adjusted to favor the detection of closely related or more distantly related proteins.

Scoring matrices

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

Figure 3.1 The BLOSUM62 scoring matrix (Henikoff and Henikoff 1992). BLOSUM62 is the most widely used scoring matrix for protein analysis and provides best coverage for general-use cases. Standard single-letter codes to the left of each row and at the top of each column specify each of the 20 amino acids. The ambiguity codes B (for asparagine or aspartic acid; Asx) and Z (for glutamine or glutamic acid; Glx) also appear, as well as an X (denoting any amino acid). Note that the matrix is a mirror image of itself with respect to the diagonal. See text for details.

20 amino acids (as well as the standard ambiguity codes).

Each value represents the logarithm of an **odds ratio** that considers how often a particular residue is observed, in nature, to replace another residue.

The **odds ratio** also considers how often a particular residue would be replaced by another if replacements occurred in a random fashion (purely by chance).

Bioinformatics, Fourth Edition. Edited by Andreas D. Baxevanis, Gary D. Bader, and David S. Wishart.

Companion Website: www.wiley.com/go/baxevanis/Bioinformatics_4e

Scoring matrices

Box 3.1 Scoring Matrices and the Log Odds Ratio

Protein scoring matrices are derived from the observed replacement frequencies of amino acids for one another. Based on these probabilities, the scoring matrices are generated by applying the following equation:

$$S_{i,j} = \log [(q_{i,j})/(p_i\, p_j)]$$

where p_i is the probability with which residue i occurs among all proteins and p_j is the probability with which residue j occurs among all proteins. The quantity $q_{i,j}$ represents how often the two amino acids i and j are seen to align with one another in multiple sequence alignments of protein families or in sequences that are known to have a biological relationship. Therefore, the log odds ratio $S_{i,j}$ (or "lod score") represents the ratio of observed vs. random frequency for the substitution of residue i by residue j. For commonly observed substitutions, $S_{i,j}$ will be greater than zero. For substitutions that occur less frequently than would be expected by chance, $S_{i,j}$ will be less than zero. If the observed frequency and the random frequency are the same, $S_{i,j}$ will be zero.

20 amino acids (as well as the standard ambiguity codes).

Each value represents the logarithm of an **odds ratio** that considers how often a particular residue is observed, in nature, to replace another residue.

The **odds ratio** also considers how often a particular residue would be replaced by another if replacements occurred in a random fashion (purely by chance).

Bioinformatics, Fourth Edition. Edited by Andreas D. Baxevanis, Gary D. Bader, and David S. Wishart.

Companion Website: www.wiley.com/go/baxevanis/Bioinformatics_4e

Scoring matrices

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

Figure 3.1 The BLOSUM62 scoring matrix (Henikoff and Henikoff 1992). BLOSUM62 is the most widely used scoring matrix for protein analysis and provides best coverage for general-use cases. Standard single-letter codes to the left of each row and at the top of each column specify each of the 20 amino acids. The ambiguity codes B (for asparagine or aspartic acid; Asx) and Z (for glutamine or glutamic acid; Glx) also appear, as well as an X (denoting any amino acid). Note that the matrix is a mirror image of itself with respect to the diagonal. See text for details.

Diagonal: score for an exact match. Some alignments are much less likely to occur in general and, in turn, are more likely to be correct.

Moving off the diagonal

- positive value implies that the substitution is observed to occur more often in an alignment than it would by chance
- negative value implies that the substitution is not observed to occur frequently and may arise more often than not by chance.

Scoring matrices

What matrices should we use them?

Table 3.1 Selecting an appropriate scoring matrix.

Matrix	Best use	Similarity
PAM40	Short alignments that are highly similar	70–90%
PAM160	Detecting members of a protein family	50–60%
PAM250	Longer alignments of more divergent sequences	~30%
BLOSUM90	Short alignments that are highly similar	70–90%
BLOSUM80	Detecting members of a protein family	50–60%
BLOSUM62	Most effective in finding all potential similarities	30–40%
BLOSUM30	Longer alignments of more divergent sequences	<30%

The Similarity column gives the range of similarities that the matrix is able to best detect (Wheeler 2003).

Bioinformatics, Fourth Edition. Edited by Andreas D. Baxevanis, Gary D. Bader, and David S. Wishart.

Companion Website: www.wiley.com/go/baxevanis/Bioinformatics_4e

Scoring matrices

- BLOSUM 62
- Let’s score an alignment:
  - No counting gaps
- HEAGAWGHE -E
- – –P– AW –HEAE
- Score: -1 + 4 + 11 + 8 + 5 + 5 = 32

Nucleotide scoring matrices

- For nucleotides, the landscape is much simpler.
- Matches/mismatches with 25% frequency (A, T, C, G).
- In some cases, we have to include ambiguities or chemical similarities.
- Gaps and gap penalties

IUPAC codes

Symbol	Bases	Origin of designation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino
K	G or T	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T	Weak interaction (2 H bonds)
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A
V	G or C or A	not-T (not-U), V follows U
D	G or A or T	not-C, D follows C
N	G or A or T or C	aNy

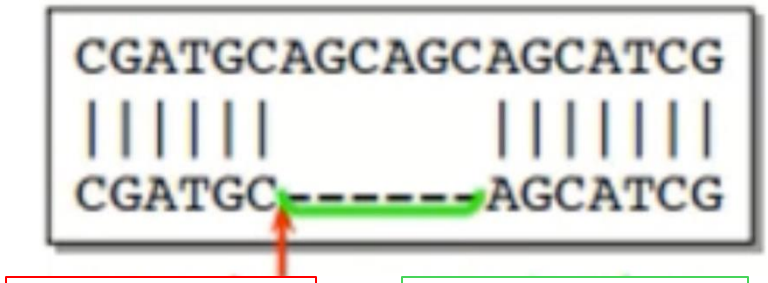
Nucleotide scoring matrices

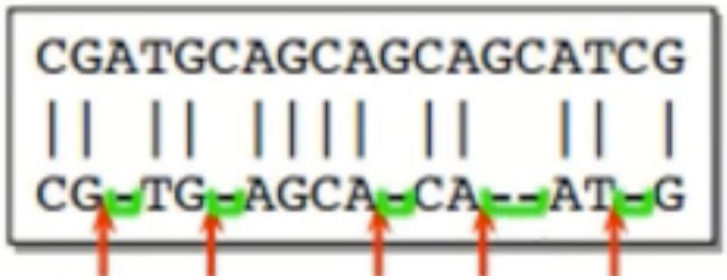
	A	T	G	C	S	W	R	Y	K	M	B	V	H	D	N
A	5	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2
T	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2
G	-4	-4	5	-4	1	-4	1	-4	1	-4	-1	-1	-4	-1	-2
C	-4	-4	-4	5	1	-4	-4	1	-4	1	-1	-1	-1	-4	-2
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-1	-1	-3	-3	-1
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-3	-3	-1	-1	-1
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-3	-1	-3	-1	-1
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-1	-3	-1	-3	-1
K	-4	1	1	-4	-2	-2	-2	-2	-1	-4	-1	-3	-3	-1	-1
M	1	-4	-4	1	-2	-2	-2	-2	-4	-1	-3	-1	-1	-3	-1
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-1	-2	-2	-2	-1
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-2	-1	-2	-2	-1
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-2	-2	-1	-2	-1
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-2	-2	-2	-1	-1
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 3.2 A nucleotide scoring table. The scoring for the four nucleotide bases is shown in the upper left of the figure, with the remaining one-letter codes specifying the IUPAC/UBMB codes for ambiguities or chemical similarities. Note that the matrix is a mirror image of itself with respect to the diagonal.

Gap penalties

- Gaps are introduced into alignments to compensate for insertions and deletions between two sequences.
- It is important to keep the number of gaps to a reasonable number: the more gaps, the less biologically plausible scenario.
- We need to have a gap scoring system. The most widely used is the **affine gap penalty**: $G + Ln$
  - G is the gap-opening penalty
  - L is the gap-extension penalty
  - n is the length of the gap





Sequence alignment representation: Dotplots

- A basic way to compare two sequences

- Allow quick identification of regions of local alignment
  - Alignments
  - Repeats (direct or inverted)
  - Insertions
  - Deletions
  - Low complexity regions

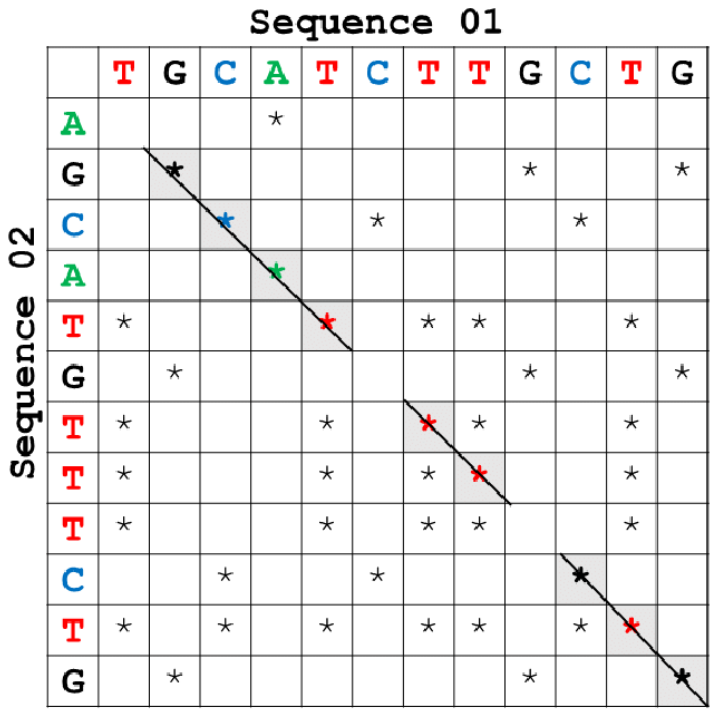


Figure modified from Junqueira et al. (2014)

The line on the diagonal will represent the regions of similarity.

Sequence alignment representation: Dotplots

- A basic way to compare two sequences

- Allow quick identification of regions of local alignment
  - Alignments
  - Repeats (direct or inverted)
  - Insertions
  - Deletions
  - Low complexity regions

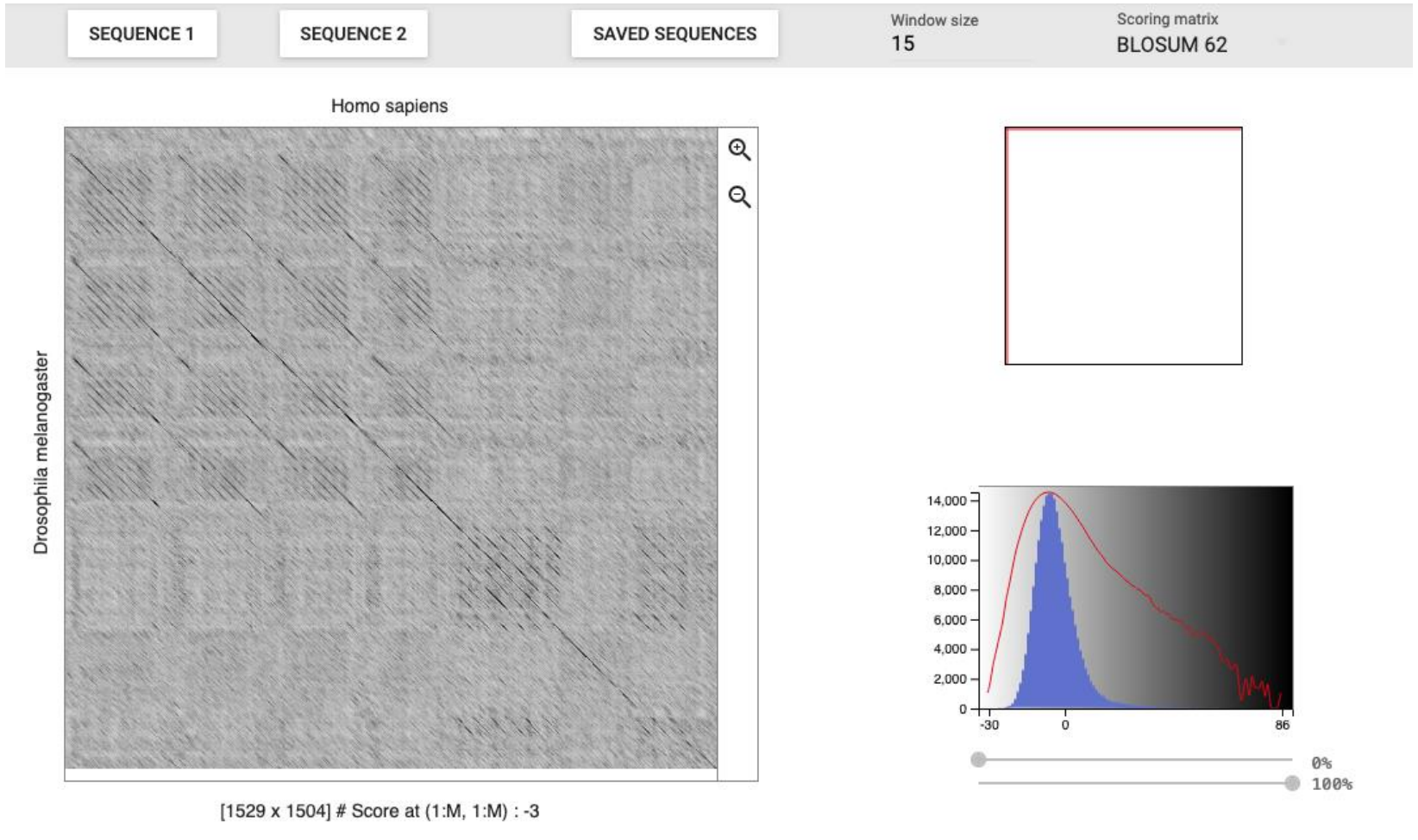
Construct a dotplot using the web-based java applet Dotlet
https://dotlet.vital-it.ch/

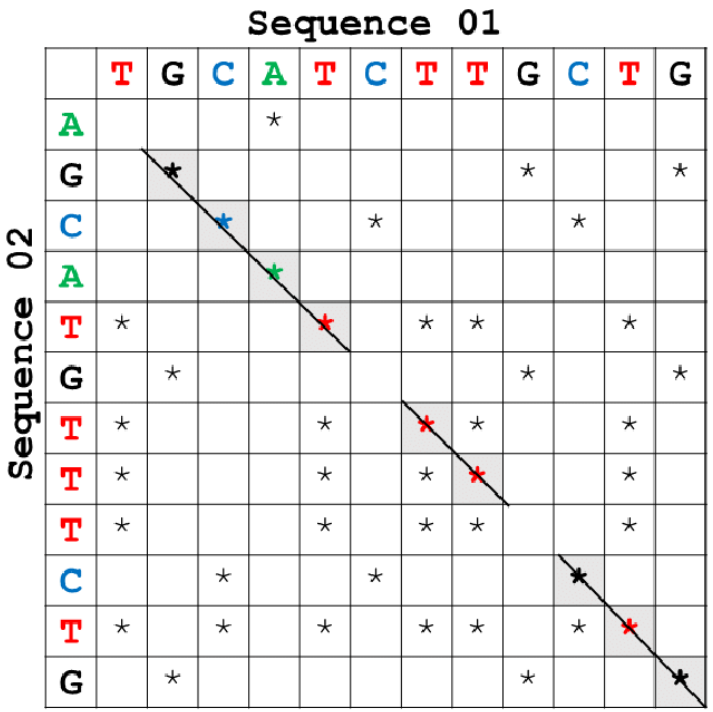
>Sequence1
TGCATCTTGCTG

>Sequence2
AGCATGTTTCTG

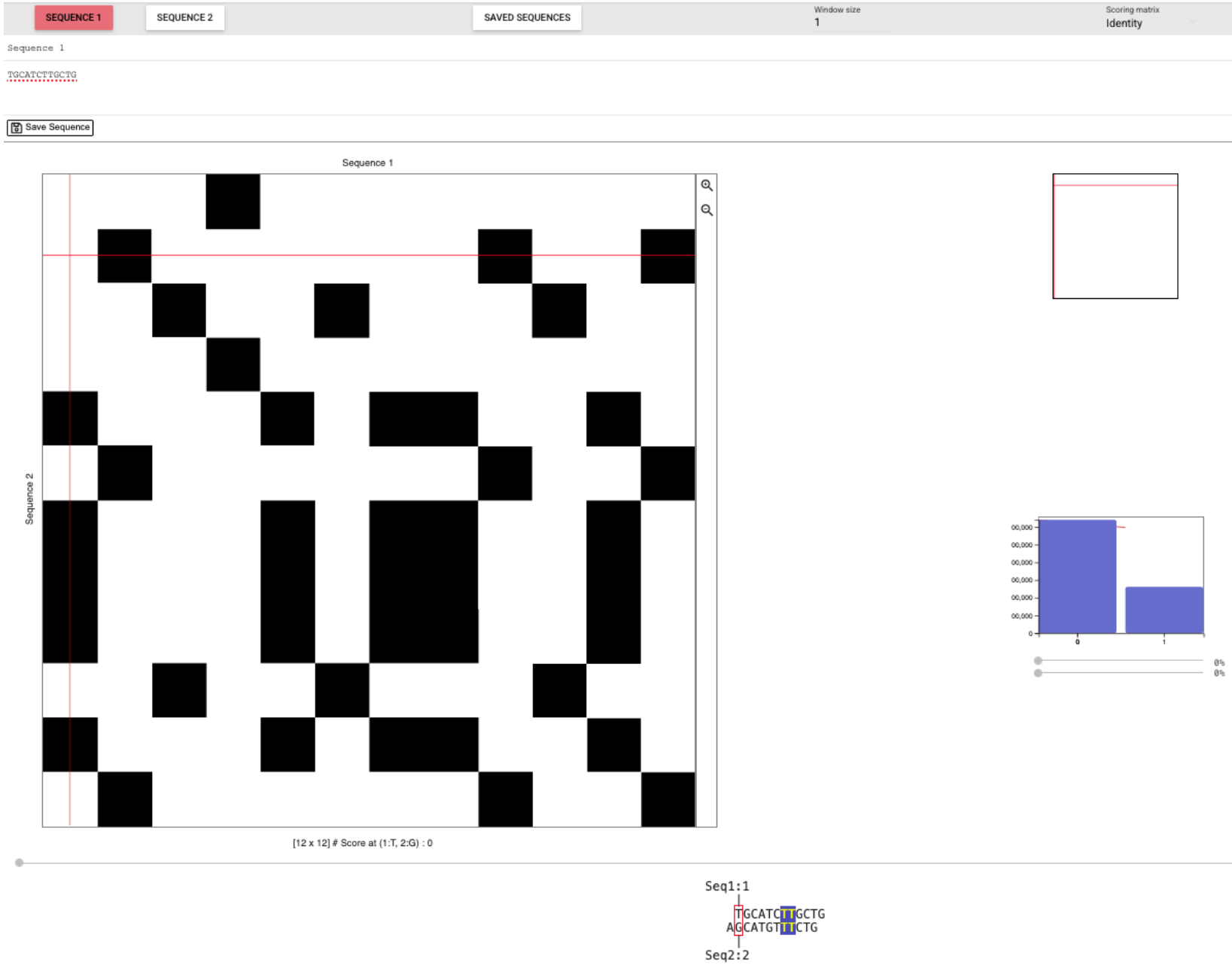
Dotlet JS *beta*

Website Initial Entry





The line on the diagonal will represent the regions of similarity.



Sequence alignment representation: Dotplots

- A basic way to compare two sequences

- Allow quick identification of regions of local alignment
  - Alignments
  - Repeats (direct or inverted)
  - Insertions
  - Deletions
  - Low complexity regions

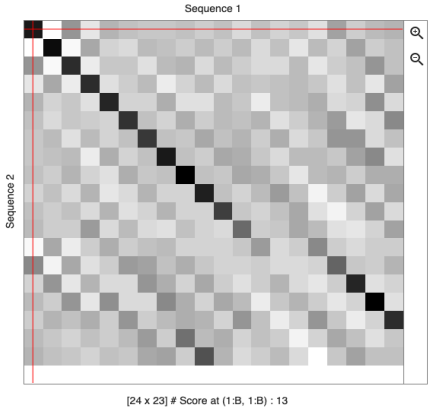
Construct a dotplot using the web-based java applet Dotlet
https://dotlet.vital-it.ch/

>Sequence1
BIQINFORMATICSALGQRITHMS

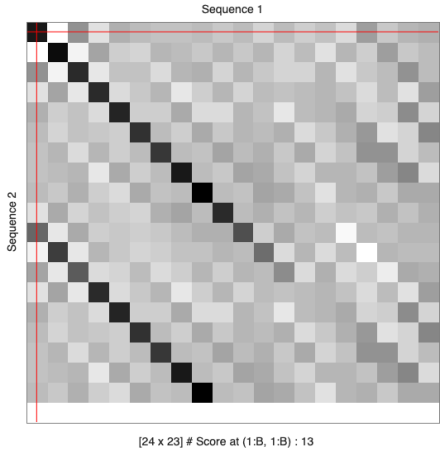
>Sequence2
BIQINFORMATICEURITHMICS

Can you build these Dotplots?

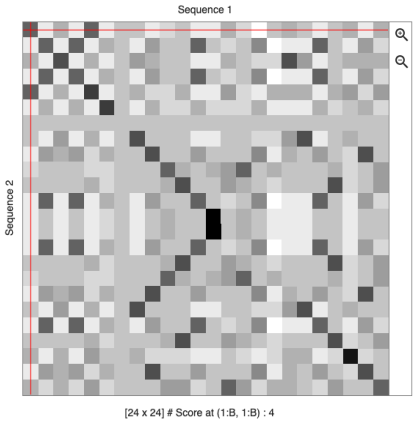
>Seq1
BIQINFORMATICSALGQRITHMS
>Seq2
???



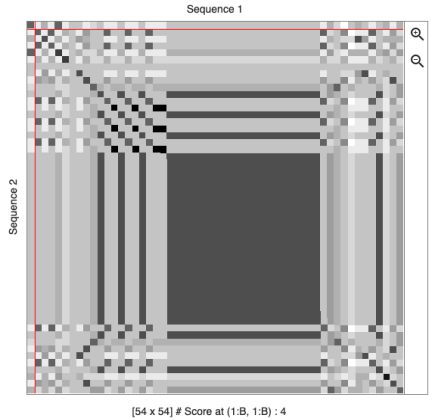
Local alignment



repeat



Inverted repeat



Low complexity

Alignment algorithms

- Now that we have a scoring system, we need to have an algorithm for finding an optimal alignment for a pair of sequences.

- There are “many” potential alignments between two seqs of length *N*.

- The **dynamic programming** algorithm is responsible for finding the optimal alignment given an additive alignment score.

- In other words, **dynamic programming** algorithms are guaranteed to find the optimal scoring alignment or set of alignments.
  *Better alignments = higher scores

  - Global alignment: Needleman-Wunsch algorithm
  - Local alignment: Smith-Waterman algorithm

Check back slide #2

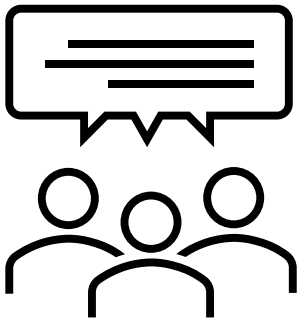
Dynamic programming algorithm

- Dynamic programming algorithms break a problem into smaller sub-problems and use the solutions of those sub-problems to construct the solution of a larger one.

- The number of sub-problems may become very large, so dynamic programming organizes computations to avoid computing values that you already know.

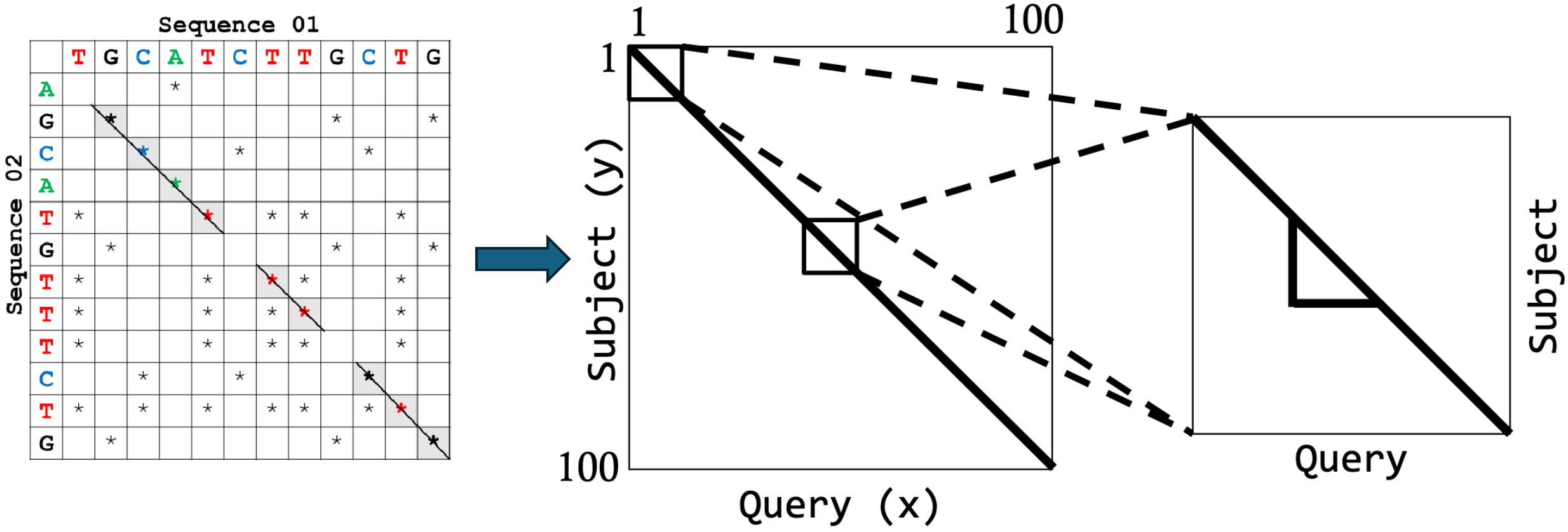
Number of possible alignments: $2^{2N}/\sqrt{2\pi N}$

N=M=16
Possible alignments?



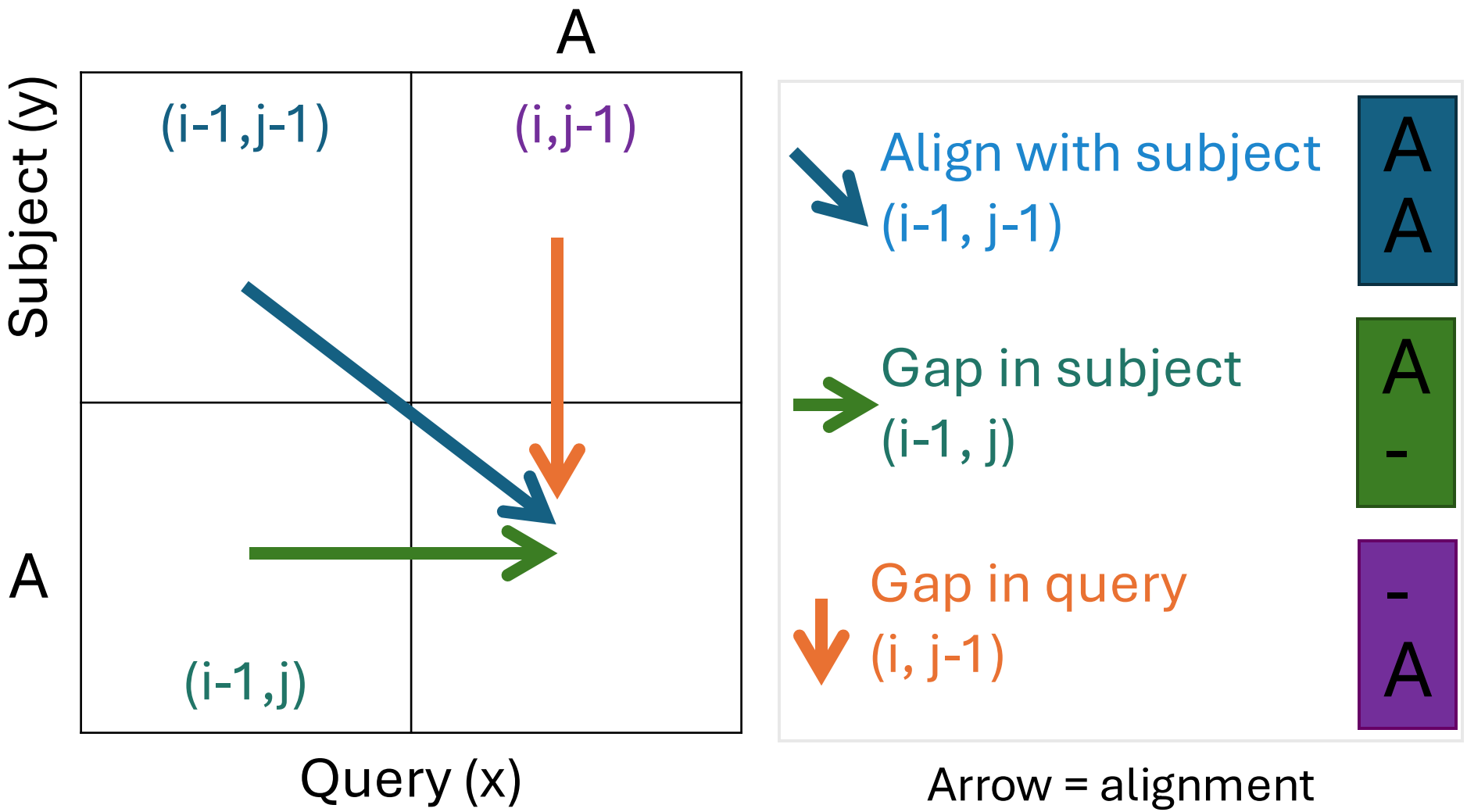
Dynamic programming algorithm

Partition the alignment problem into smaller subproblems



Assume the query and subject sequences are the same

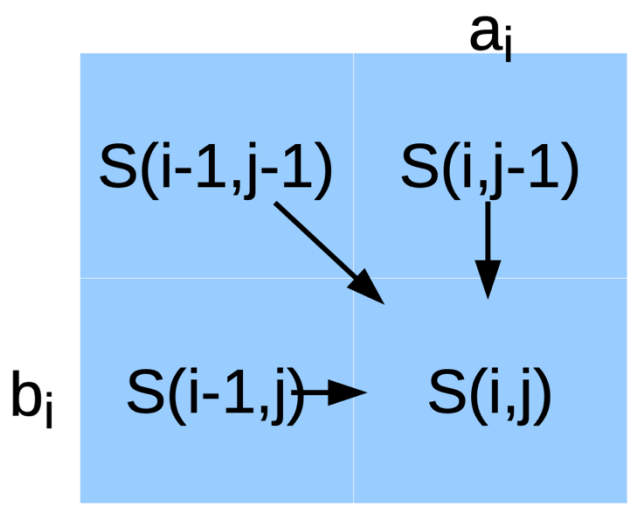
Three different ways to reach cell (i,j) in the alignment matrix



Dynamic programming algorithm

There are three different ways to reach each cell (i,j) in the alignment matrix

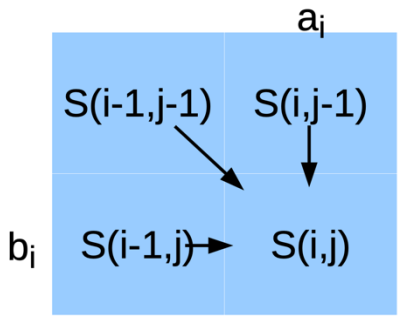
Subproblem scoring alignments



$$S(i,j) = \max \begin{cases} S(i-1,j-1) + score(a_j, b_j) \\ S(i,j-1) + gap \\ S(i-1,j) + gap \end{cases}$$

Dynamic programming algorithm

- Global alignment: Needleman-Wunsch algorithm



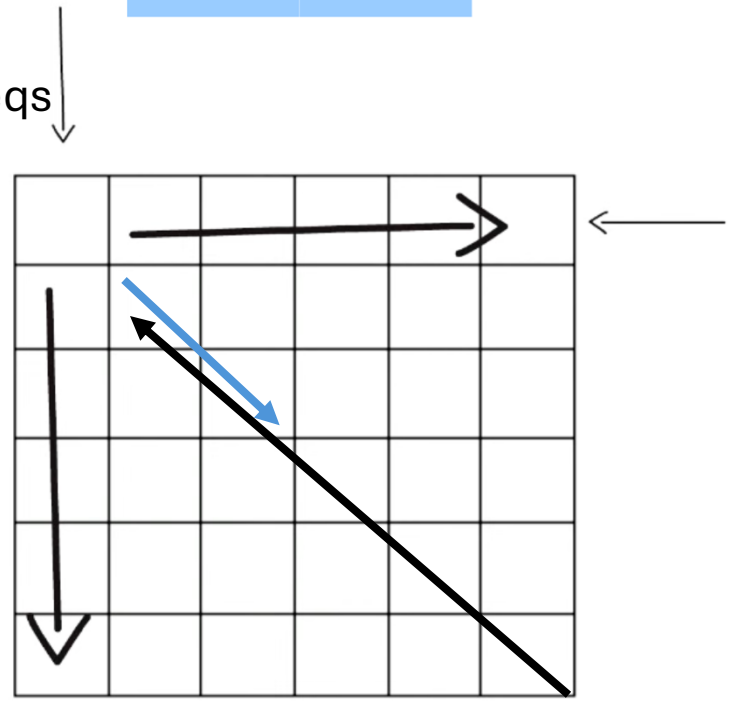
Steps:

1. Initialization

   Create a matrix with M + 1 columns and N + 1 rows (M and N = seqs sizes)

2. Matrix fill (scoring)
   - Fill with the highest possible score
   - Align with diagonal (next position)
   - Align off-diagonal (insertion/gaps)

3. Traceback (alignment)
   - Move from the last corner and follow (back) the arrows



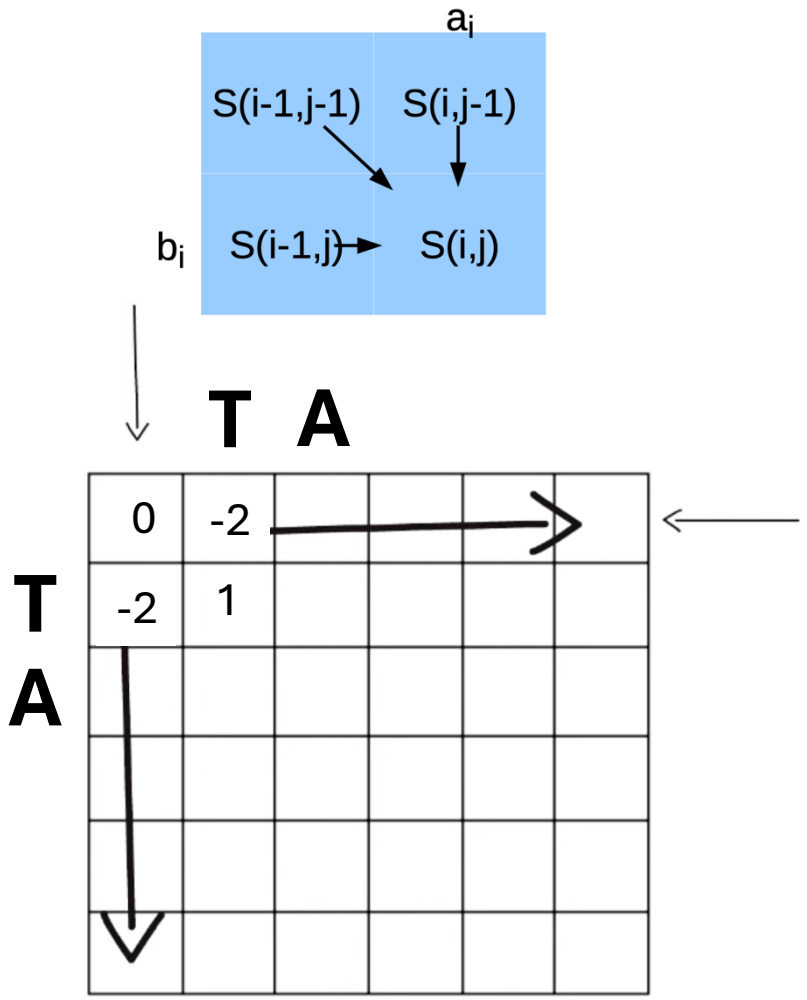
Dynamic programming algorithm

- Global alignment: Needleman-Wunsch algorithm

- Fill 1st block with zero
- Fill 1st row and 1st column with gap penalty multiples

- While filling the matrix:
  - Horizontal: score + gap penalty
  - Vertical: score + gap penalty
  - Diagonal: score + match/mismatch

Let's assume the Score Matrix:
- Match = 1
- Mismatch = -1
- Gap penalty = -2



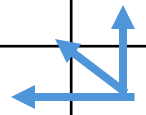
Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm

Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

		T	A	T	G	A
	0	-2	-4	-6	-8	-10
T	-2	1				
A	-4					
C	-6					
G	-8					
A	-10					



Pick highest value

Top value:
(-2)+(-2)=-4
Left value:
(-2)+(-2)=-4
Diagonal value:
0+1 (T-T)=**1**

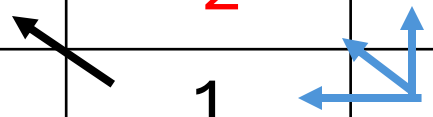
Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm

Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

		T	A	T	G	A
	0	-2	-4	-6	-8	-10
T	-2	1	-1			
A	-4					
C	-6					
G	-8					
A	-10					

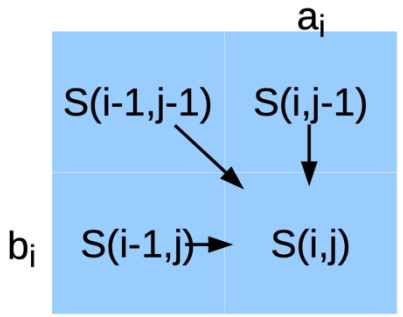


Pick highest value

Top value:
(-4)+(-2)=-6
Left value:
1+(-2)=**-1**
Diagonal value:
(-2)+(-1) (A-T)=-3

Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm



Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

		T	A	T	G	A
	0	-2	-4	-6	-8	-10
T	-2	1 (1, -4, -4)	-1 (-4, -3, -1)	-3 (-6)	-5	-7
A	-4					
C	-6					
G	-8					
A	-10					

Pick highest value

Top value:
(-4)+(-2)=-6
Left value:
1+(-2)=**-1**
Diagonal value:
(-2)+(-1) (A-T)=-3

Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm

Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

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

Pick highest value

Top value:

Left value:

Diagonal value:

Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm

Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

Traceback (alignment)

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

Dynamic programming algorithm

Global alignment: Needleman-Wunsch algorithm

Let's assume the Score Matrix:

- Match = 1
- Mismatch = -1
- Gap penalty = -2

Traceback (alignment)

We move from last cell (lower corner) and follow arrow from which cell the current value (cell) come from.

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

Alignments:

```
T A  T G A
|  |  .  |  |
T A C G A
```

Dynamic programming algorithm

Want to know more?

- ➢ Global alignment: Needleman-Wunsch algorithm

- ❑ What is dynamic programming? Durbin 2004. (Perusal)

- ❑ Protein alignment (BLOSUM50 score matrix): Durbin book (chapter 2.3)

- ❑ Freiburg RNA Tools: http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Needleman-Wunsch

Dynamic programming algorithm

	H	E	A	G	A	W	G	H	E	E
P	−2	−1	−1	−2	−1	−4	−2	−2	−1	−1
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
W	−3	−3	−3	−3	−3	**15**	−3	−3	−3	−3
H	**10**	0	−2	−2	−2	−3	−2	**10**	0	0
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**

Figure 2.3 *The two example sequences we will use for illustrating dynamic programming alignment algorithms, arranged to show a matrix of corresponding* BLOSUM50 *values per aligned residue pair. Positive scores are in bold.*

Seq1
HEAGAWGHEE

Seq2
PAWHEAE

- Gap penalty = -8

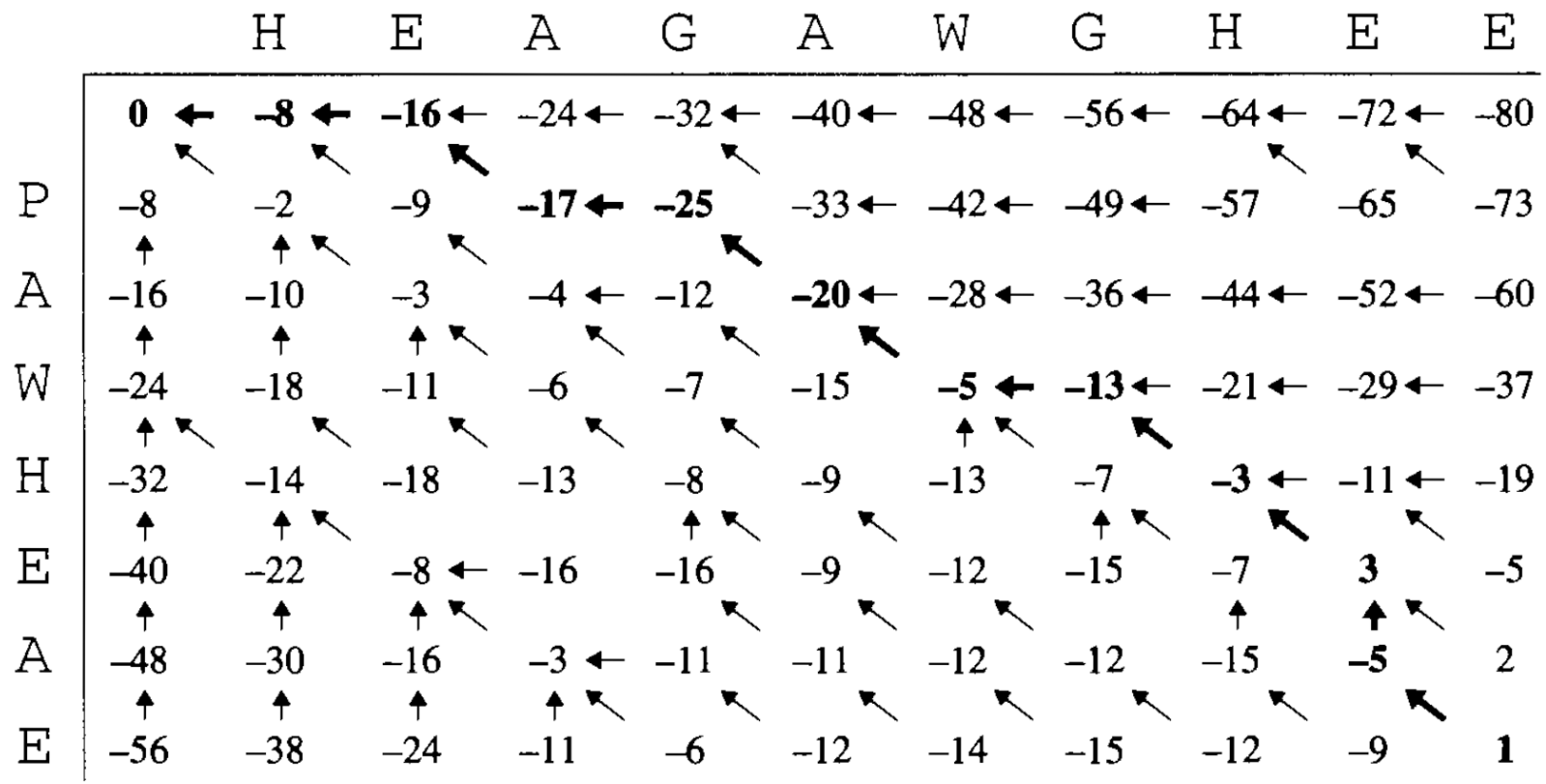
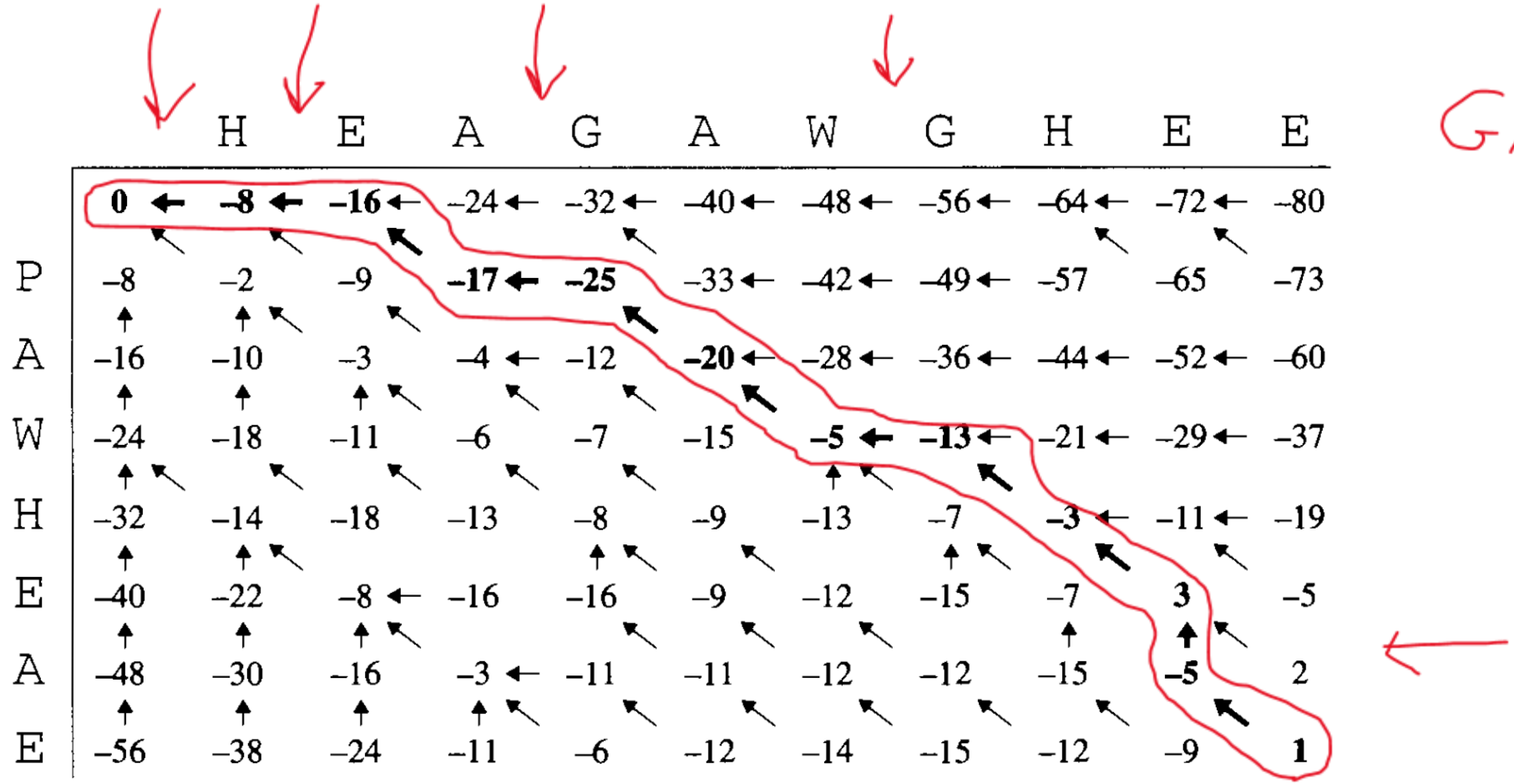


Figure 2.5 *Above, the global dynamic programming matrix for our example sequences, with arrows indicating traceback pointers; values on the optimal alignment path are shown in bold. Below, a corresponding optimal alignment, which has total score* 1.




```
HEAGAWGHE-E
--P-AW-HEAE
```

Figure 2.5 *Above, the global dynamic programming matrix for our example sequences, with arrows indicating traceback pointers; values on the optimal alignment path are shown in bold. Below, a corresponding optimal alignment, which has total score* 1.

Alignments algorithms
- Lecture02 & Lecture03 -

- Pairwise Alignment
  - Global
  - Local
  - Scoring system

- Multiple Sequence Alignment (MSA)

- Heuristic – Database search