BIOINFORMATICS 1

or why all biologists need computers

Wojciech Makałowski
Institute of Bioinformatics
Faculty of Medicine

INTRODUCTION TO SEQUENCE ANALYSIS

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE
THISISANMNCESTRALSEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE
THISISANMNCESTRALSEQUENCE
THISISANMNCESTRAWSEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE
THISISANMNCESTRALSEQUENCE
THISISANMNCESTRAWSEQUENCE
THISISANMPCESTRAWSEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE
THISISANMNCESTRALSEQUENCE
THISISANMNCESTRAWSEQUENCE
THISISANMPCESTRAWSEQUENCE
THISISCNMPESTRAWSEQUENCE

Please note deletion of “C”

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCNMPESTRAWSEQUENCE

Gene duplication or speciation!

THISISCNMPESTRAWSEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCNMPESTRAWSEQUENCE
THISISCOMPEETRAWSEQUENCE

THISISCNMPESTRAWSEQUENCE
THISISNMPERSXTRASEQUENCE

Please note deletion of “C” and “W”

compensated by insertion of “R” and “X”

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPEETLAWSEQUENCE

THISISCNMPEEXTRASEQUENCE

Please note insertion of “C”

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPLETLNAWSEQUENCE

THISISCSMPEEXTRASEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPLETLNAWSEQUENCE

THISISCSUPEEXTRASEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPLETLNEWSEQUENCE

THISISCSUPEEXTRASEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPLETELYNEWSEQUENCE

THISISSUPEREXTRASEQUENCE

Please note another deletion of “C” and insertion of “R”

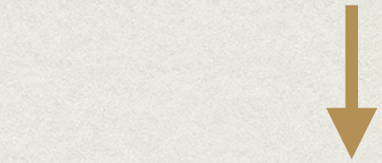
EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISCOMPLETELYNEWSEQUENCE

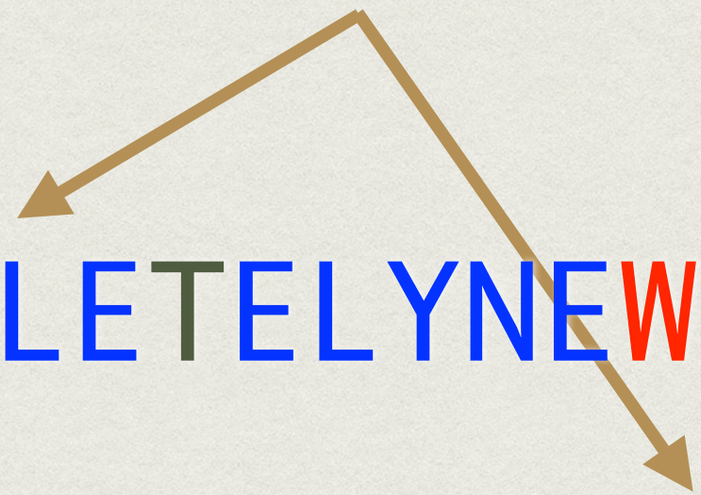
THISISSUPEREXTRASEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE



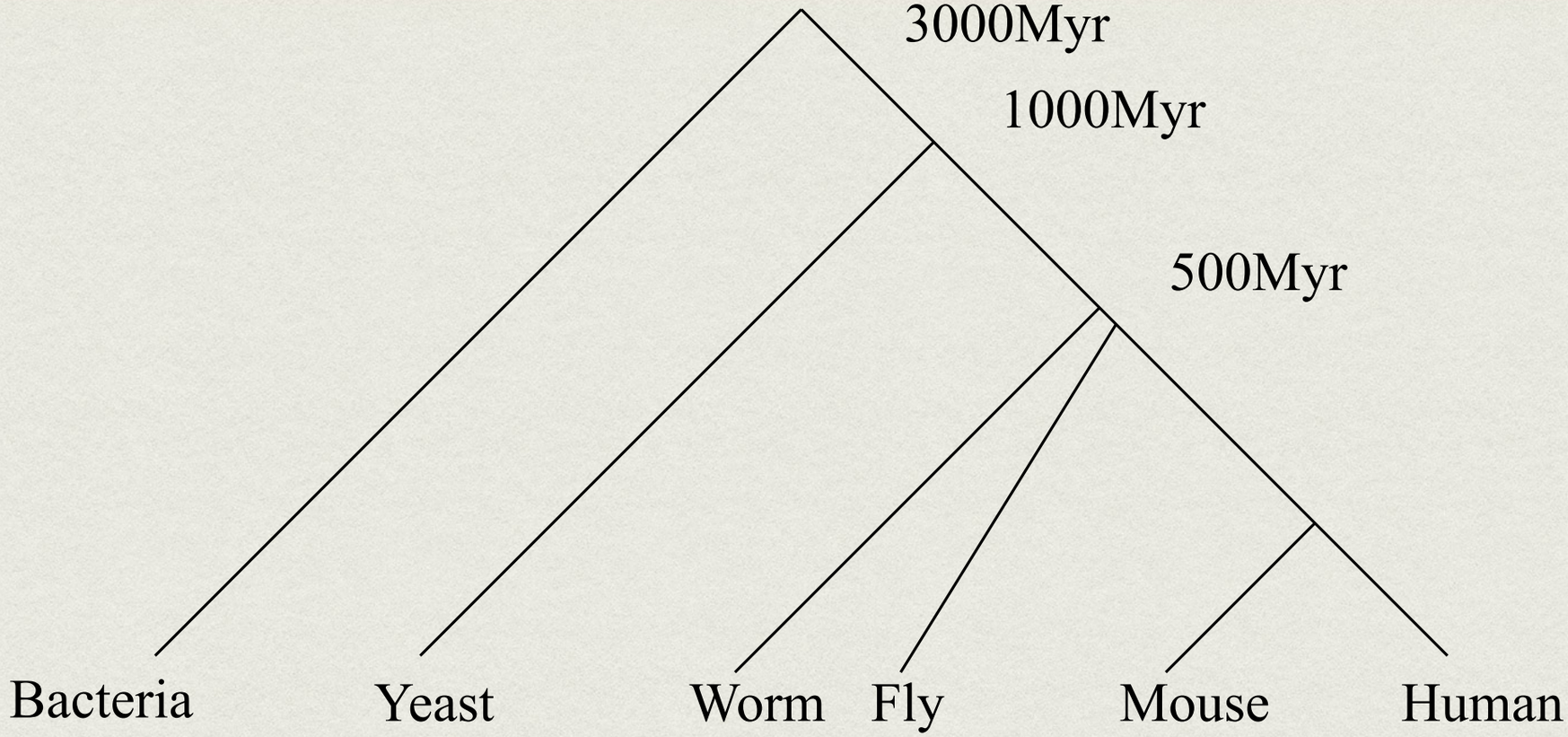
THISISCNMPESTRAWSEQUENCE



THISISCOMPLETELYNEWSEQUENCE

THISISSUPEREXTRASEQUENCE

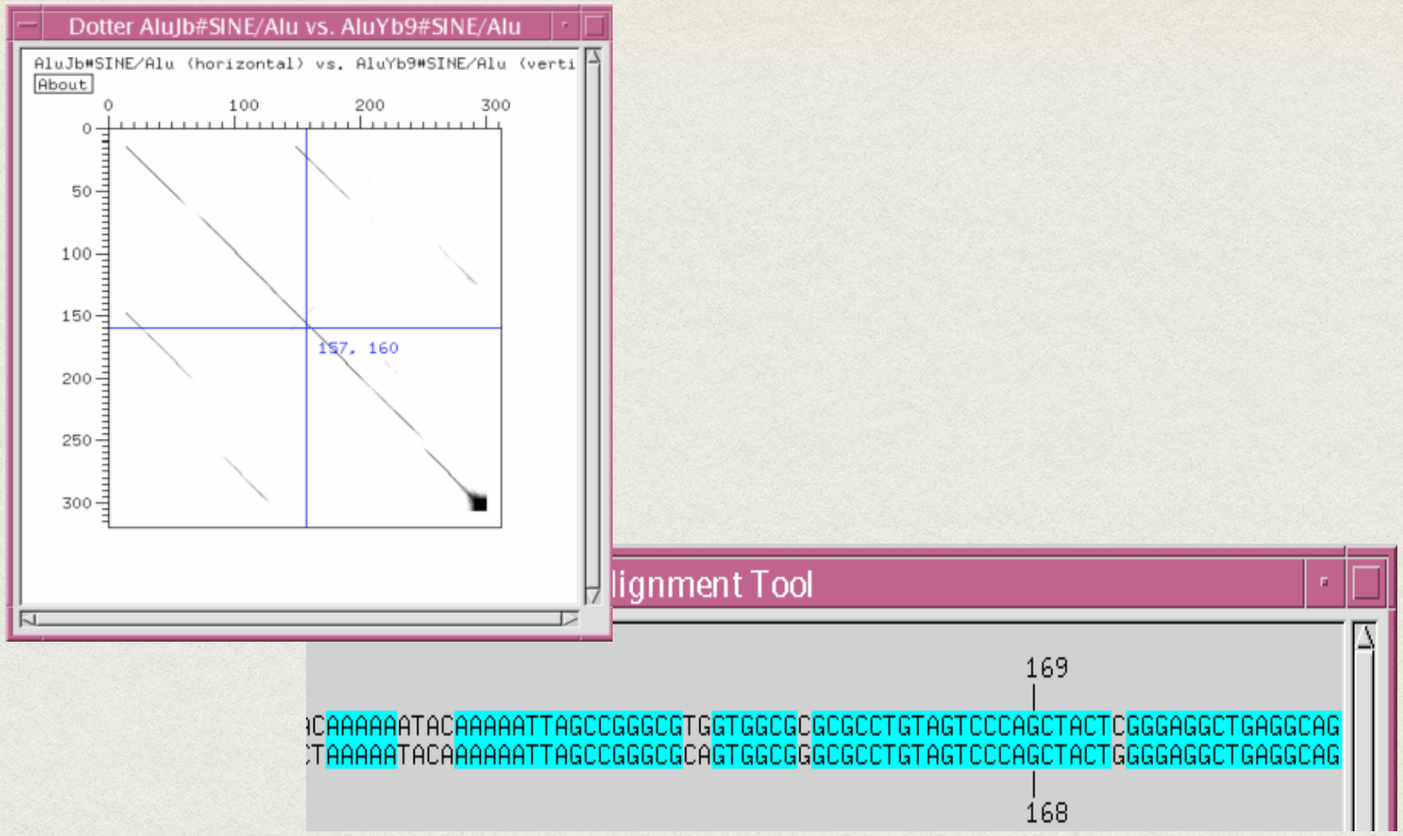
HUMAN COLON CANCER GENE AND BACTERIAL DNA REPAIR GENE



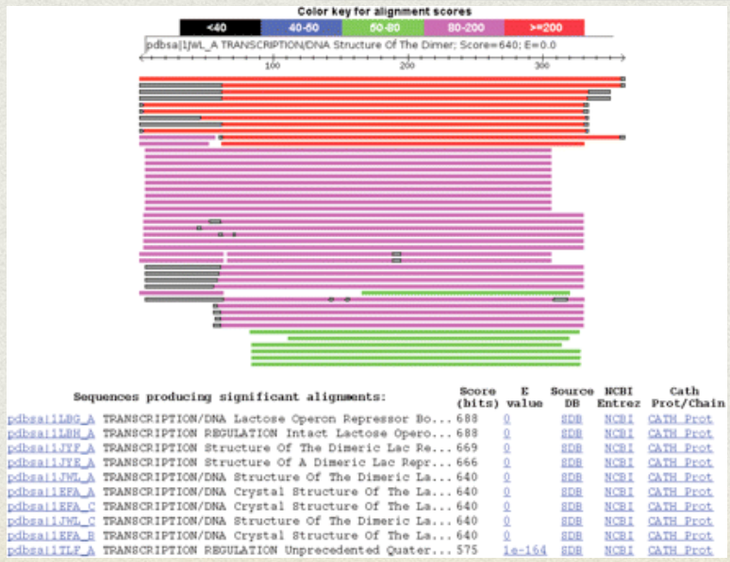

```
MSH2_Human TGVIVLMAQIGCFVPCESAEVSIVDCILARVGAGDSQLKGVSTFMAEMLETASILRSATK
SPE1_DROME VGTAVLMAHIGAFVPCSLATISMVDSILGRVGASDNIIKGLSTFMVEMIETSGIIRTATD
MSH2_Yeast VGVISLMAQIGCFVPCEEAEIAIVDAILCRVGAGDSQLKGVSTFMVEILETASILKNASK
MUTS_ECOLI TALIALMAYIGSYVPAQKVEIGPIDRIFTRVGAADDLASGRSTFMVEMTETANILRNATE
                  *** **  **          * *  ****          **** *  **  *   *
```

MAJOR TECHNIQUES TO BE DISCUSSED

- Dot Matrix plots



- Sequence alignments



- Similarity searches

HOW TO SOLVE THE PROBLEM - HUMAN OR COMPUTER?

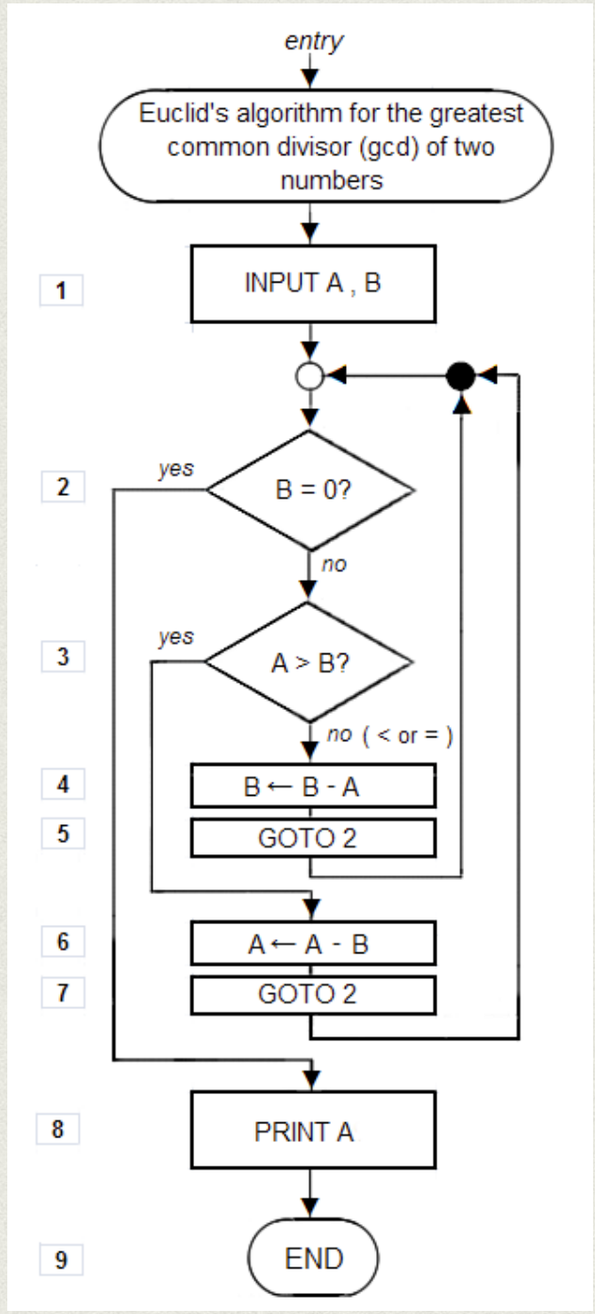
- very smart
- slow
- error prone
- doesn't like repetitive tasks

- not so smart (stupid)
- extremely fast
- very accurate
- doesn't understand human languages;

needs instruction provided in a special way

ALGORITHM

A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.



Dot Matrix Plots

DOT MATRIX PLOTS

- Sensitive qualitative indicators of similarity
- Better than alignments in some ways
  - rearrangements
  - repeated sequences
- Rely on visual perception (not quantitative)
- Useful for RNA structure determination

DOT MATRIX PLOTS

- Simplest method - put a dot wherever sequences are identical
- A little better - use a scoring table, put a dot wherever the residues have better than a certain score (especially useful for amino acid sequence comparison)
- Or, put a dot wherever you get at least n matches in a row (identity matching, compare/word)
- Even better - filter the plot

WINDOWED SCORES ALGORITHM

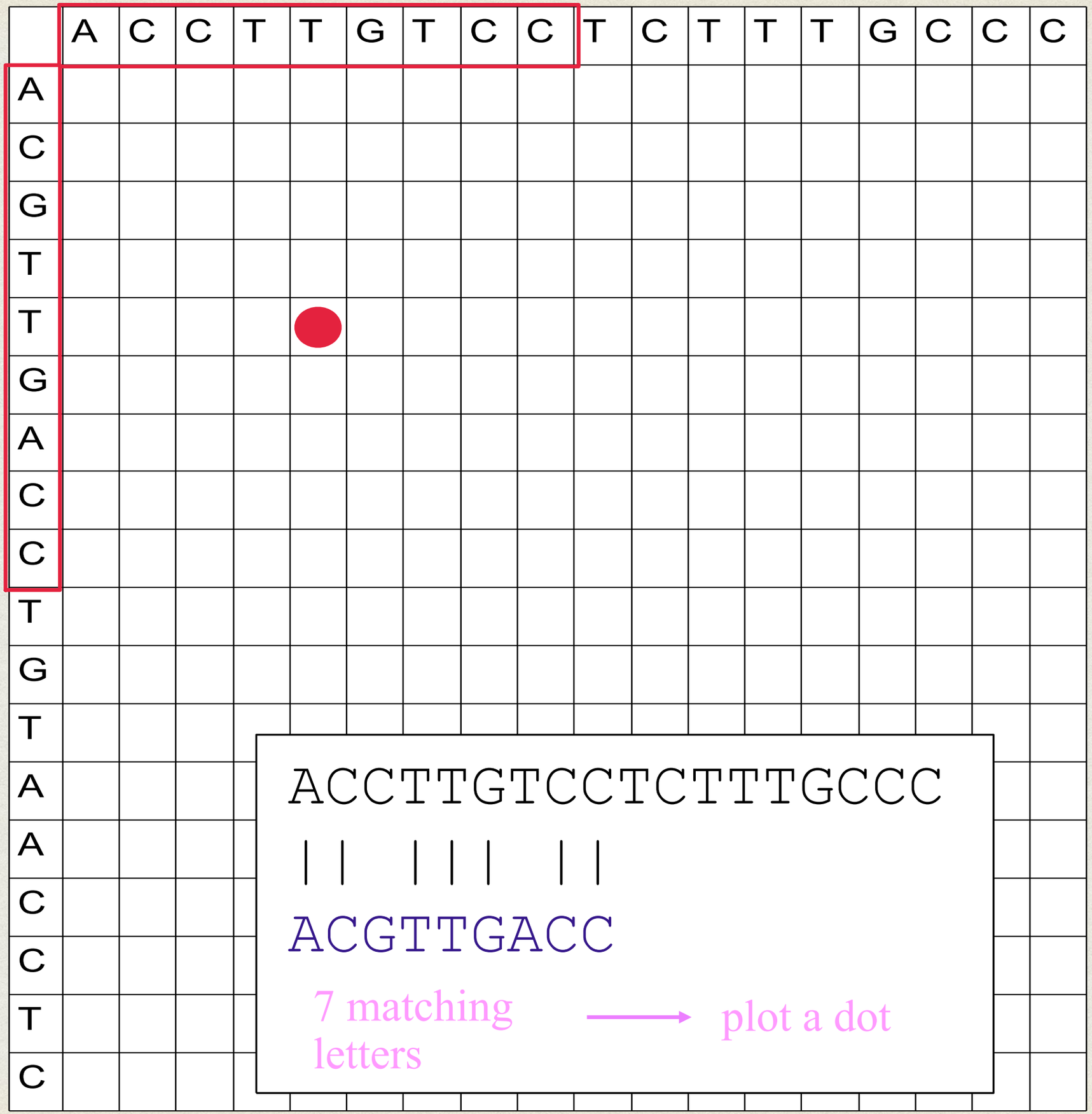
1. calculate a score within a window of a given size, for example six
2. plot a point if score is over a threshold (stringency), for example 70%
3. move the window over a given step, for example one
4. repeat step one to three till the end of sequence

WINDOWED SCORES EXAMPLE

Let's compare two nucleotide sequences

ACCTTGTCCTCTTTGCCC
ACGTTGACCTGTAACCTC

using following parameters:
window size = 9, step = 3, threshold = 4



window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●			?										
G																		
A																		
C																		
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| |

ACGTTGACC

3 matching letters → no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●						?							
G																		
A																		
C																		
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| |

ACGTTGACC

2 matching letters ⟶ no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C																		
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

```
ACCTTGTCCTCTTTGCCC
  |  |||  ||
ACGTTGACC
```

6 matching letters → plot a dot

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C					?													
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

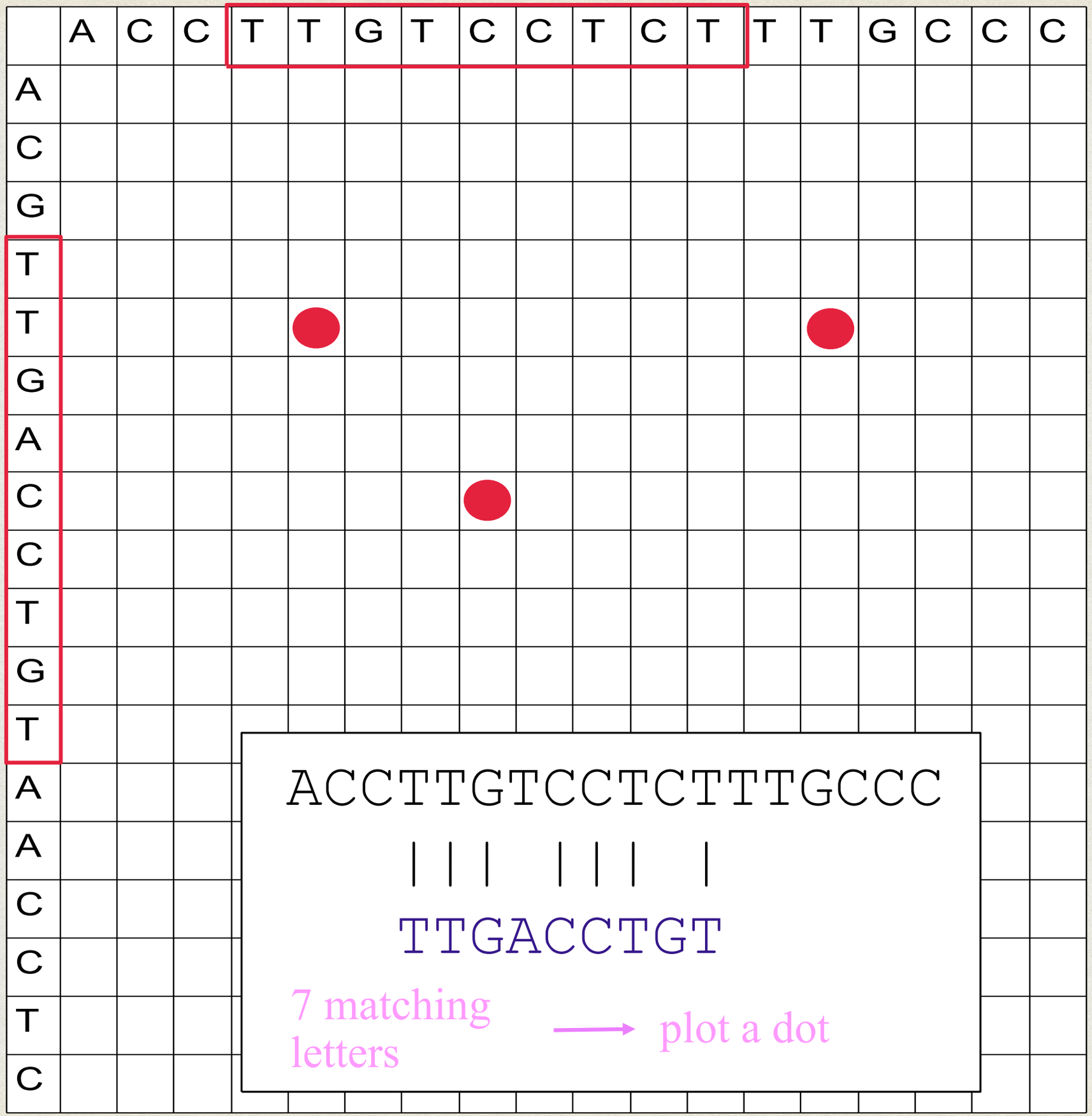
ACCTTGTCCTCTTTGCCC

|

TTGACCTGT

1 matching letter → no action

window size = 9
step = 3
threshold = 4



window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●			?							
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| | |

TTGACCTGT

3 matching letters → no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●						?				
C																		
T																		
G																		
T																		
A																		
A																		
C																		
C																		
T																		
C																		

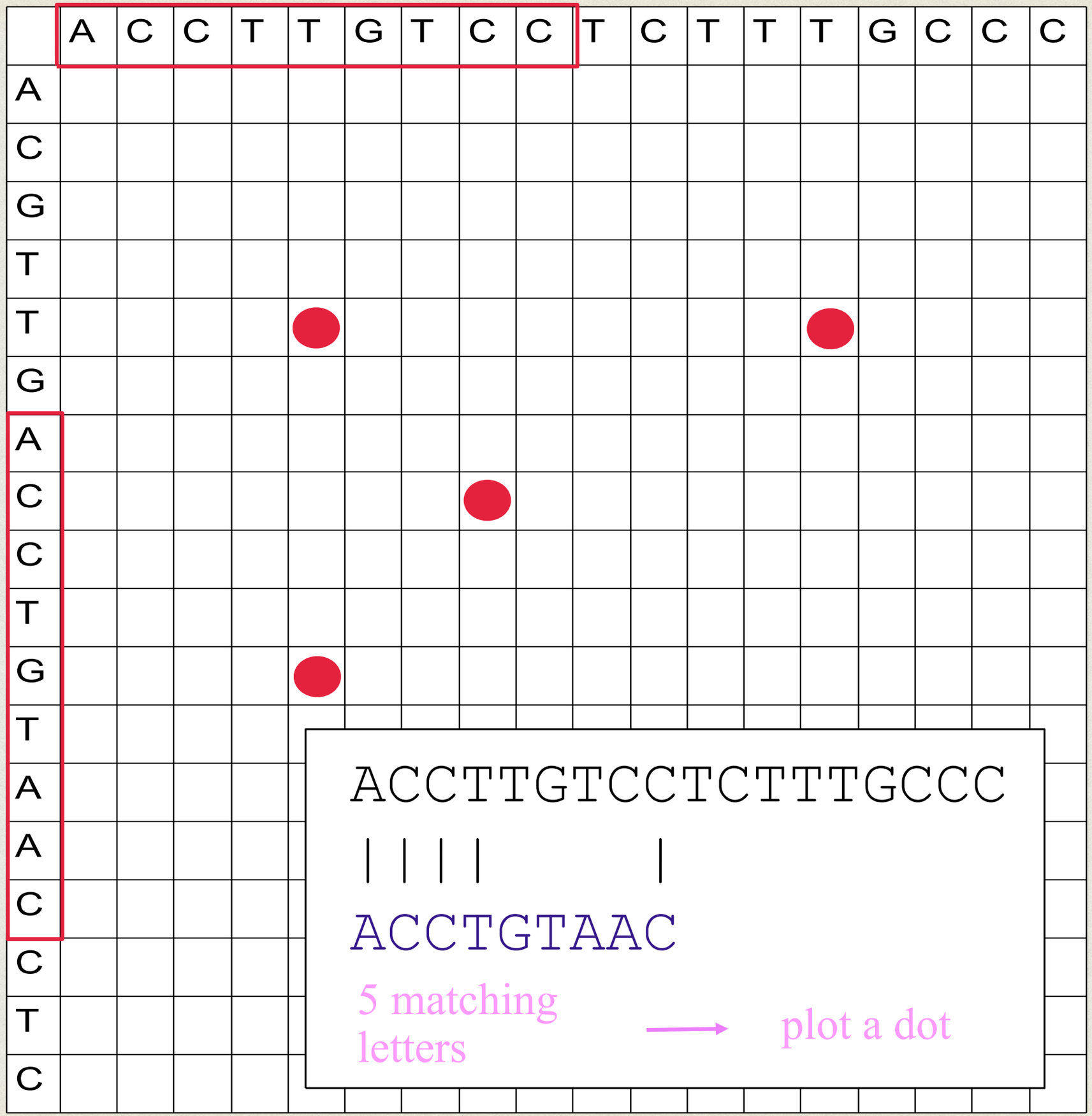
ACCTTGTCCTCTTTGCCC

|

TTGACCTGT

1 matching letter → no action

window size = 9
step = 3
threshold = 4



window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●										
C																		
T																		
G					●			?										
T																		
A																		
A																		
C																		
C																		
T																		
C																		

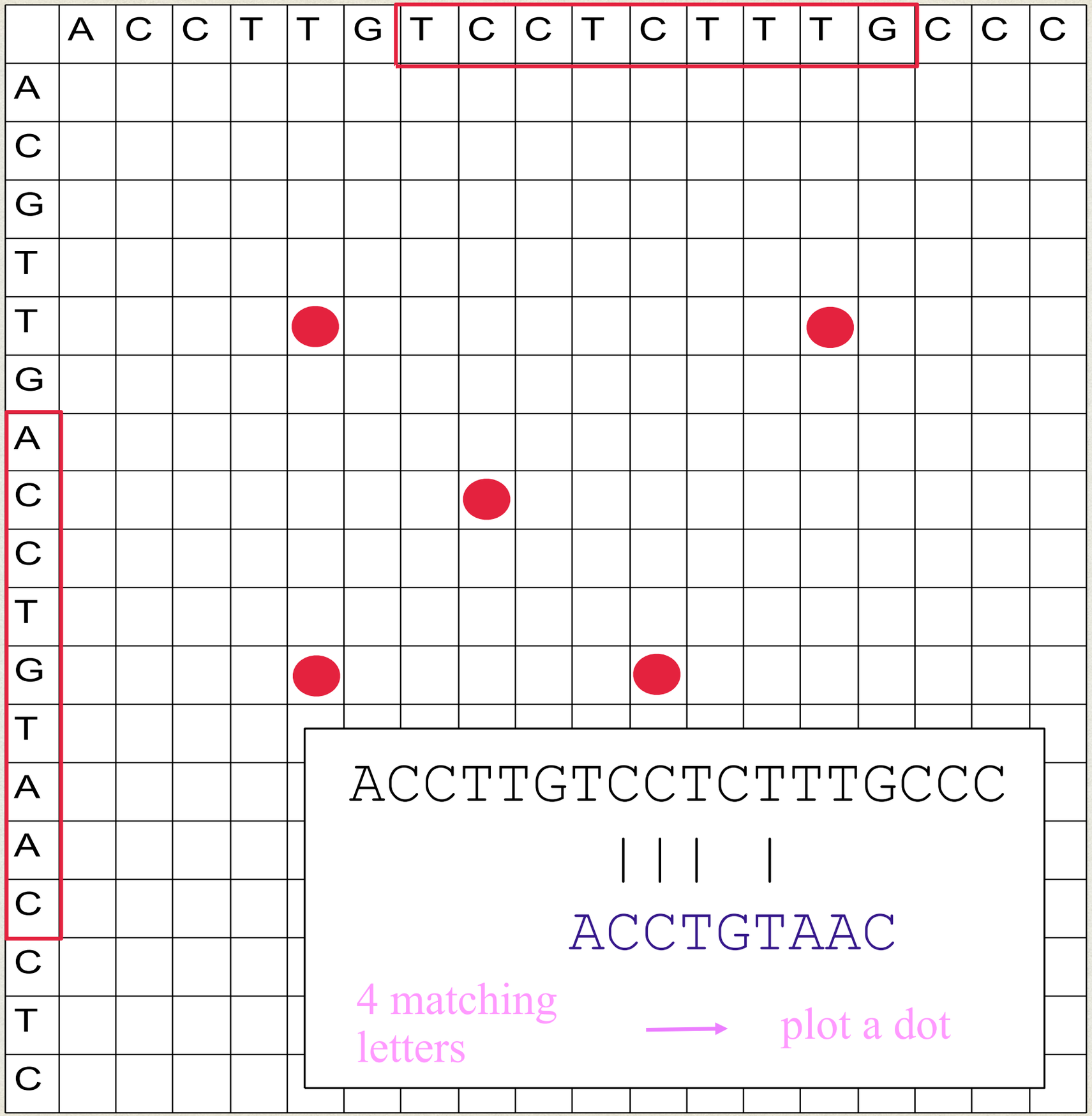
ACCTTGTCCTCTTTGCCC

|

ACCTGTAAC

1 matching letter → no action

window size = 9
step = 3
threshold = 4



window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●										
C																		
T																		
G					●						●			?				
T																		
A																		
A																		
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| | |

ACCTGTAAC

3 matching letters → no action

window size = 9
step = 3
threshold = 4

ACCTTGTCCTCTTTGCCC

|

TGTAACCTC

1 matching letter → no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
T																		
G																		
T																		
A																		
A					?													
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| |

TGTAACCTC

2 matching letters → no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
T																		
G																		
T																		
A																		
A								?										
C																		
C																		
T																		
C																		

ACCTTGTCCTCTTTGCCC

| |

TGTAACCTC

2 matching letters → no action

window size = 9
step = 3
threshold = 4

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
T																		
G																		
T																		
A																		
A											?							
C																		
C																		
T																		
C																		

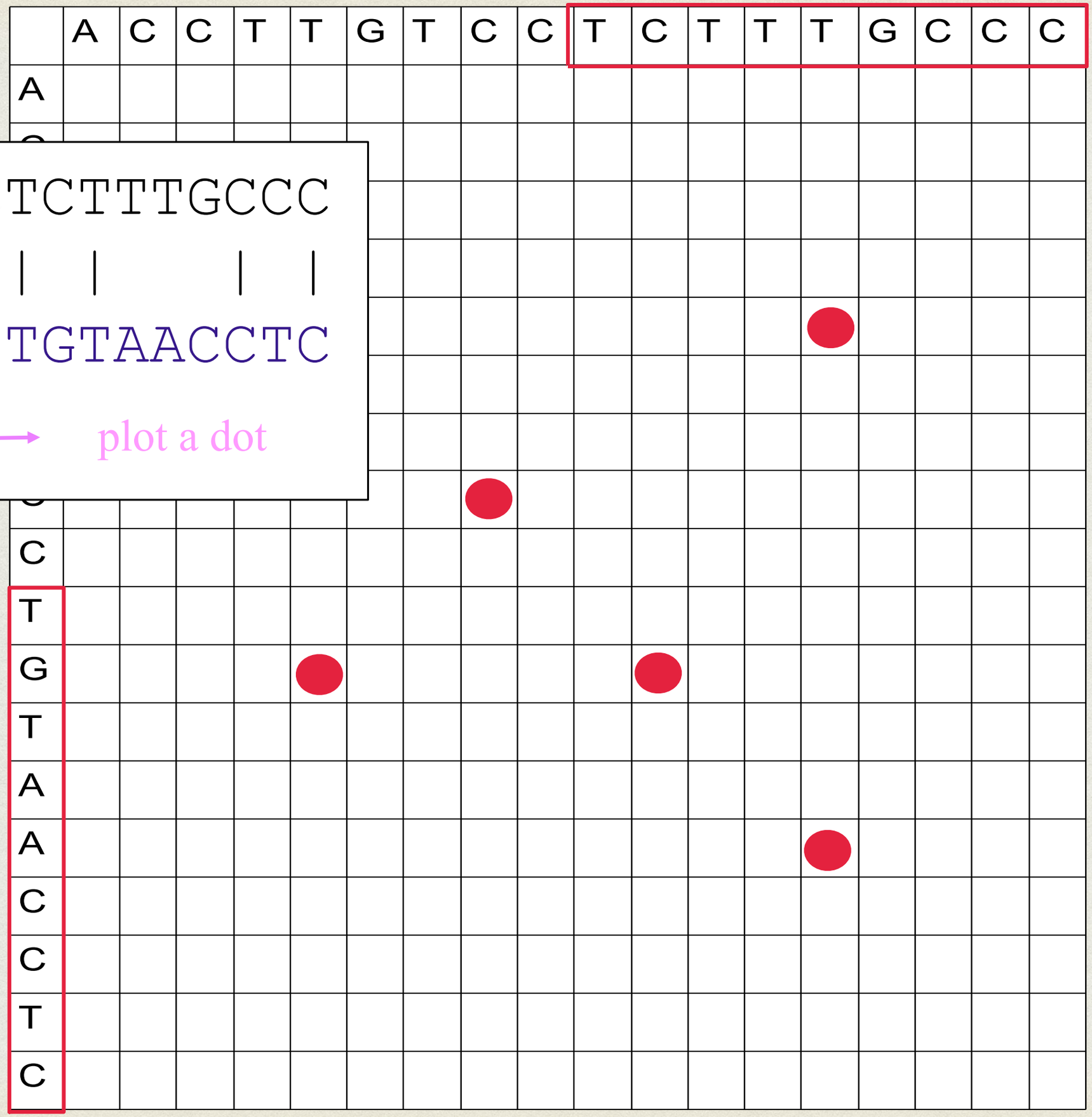
ACCTTGTCCTCTTTGCCC

| | | | |

TGTAACCTC

4 matching letters → plot a dot

window size = 9
step = 3
threshold = 4



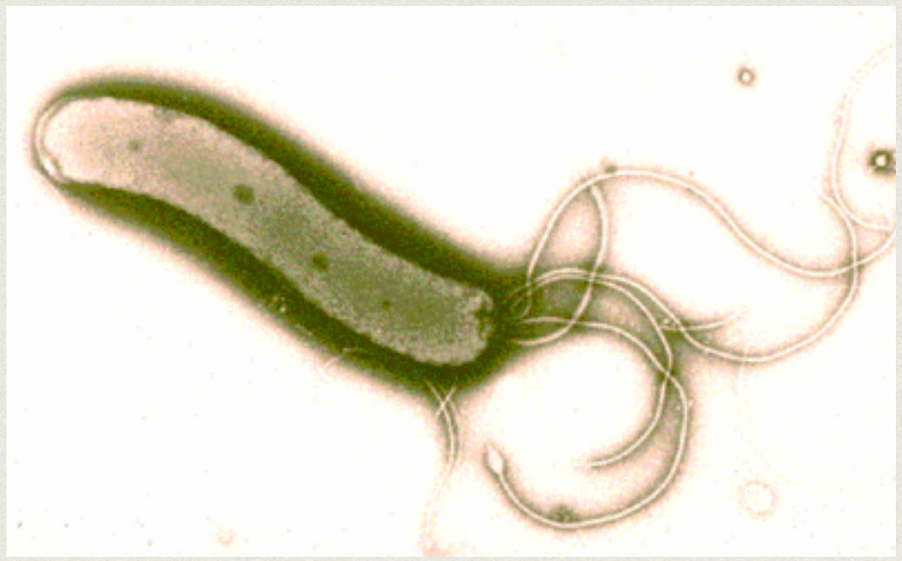
	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●										
C																		
T																		
G					●						●							
T																		
A																		
A														●				
C																		
C																		
T																		
C																		

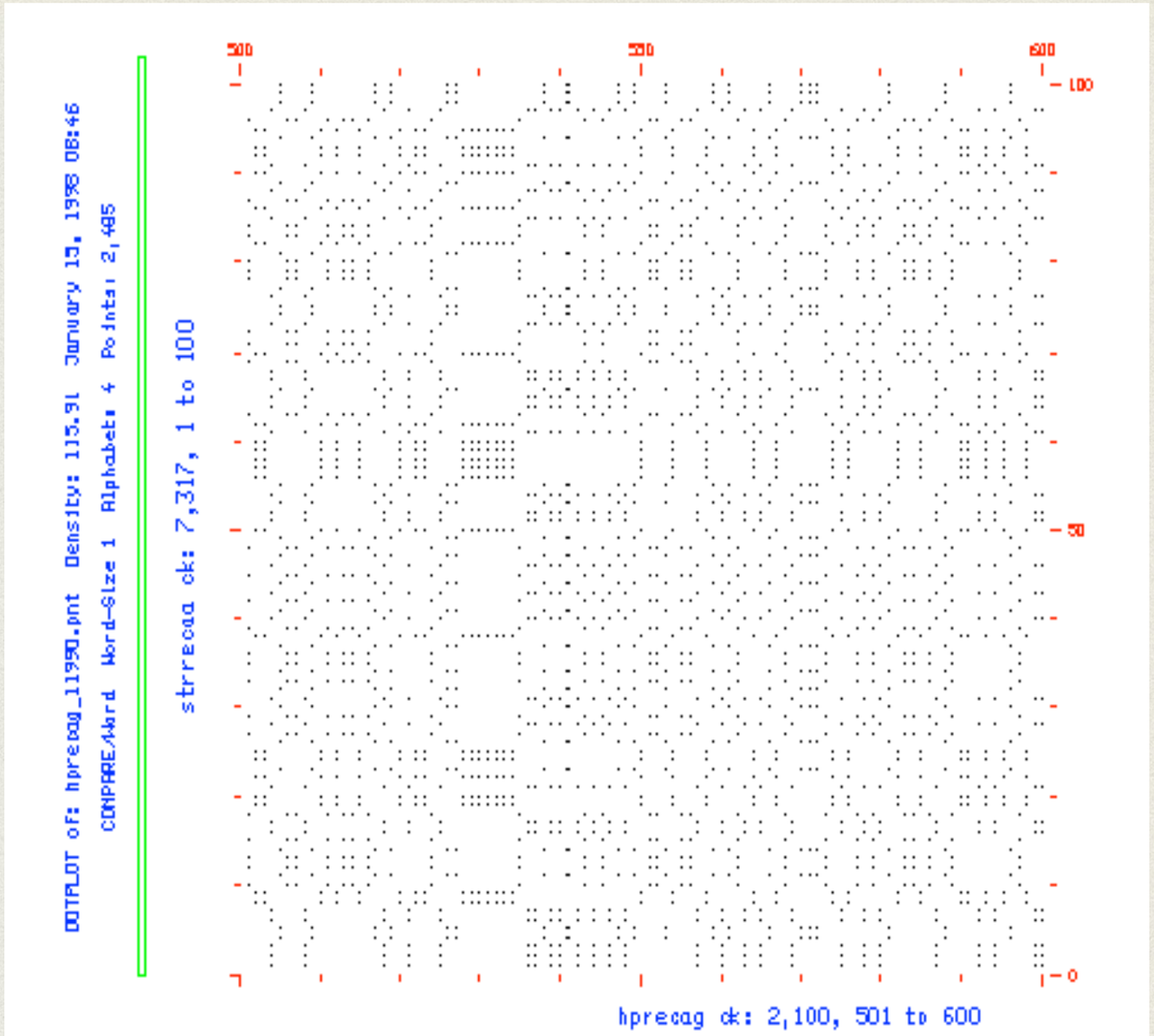
For similar sequences dots form a diagonal line

	A	C	C	T	T	G	T	C	C	T	C	T	T	T	G	C	C	C
A																		
C																		
G																		
T																		
T					●									●				
G																		
A																		
C								●										
C																		
T																		
G					●						●							
T																		
A																		
A														●				
C																		
C																		
T																		
C																		

DOT PLOT - EXAMPLES

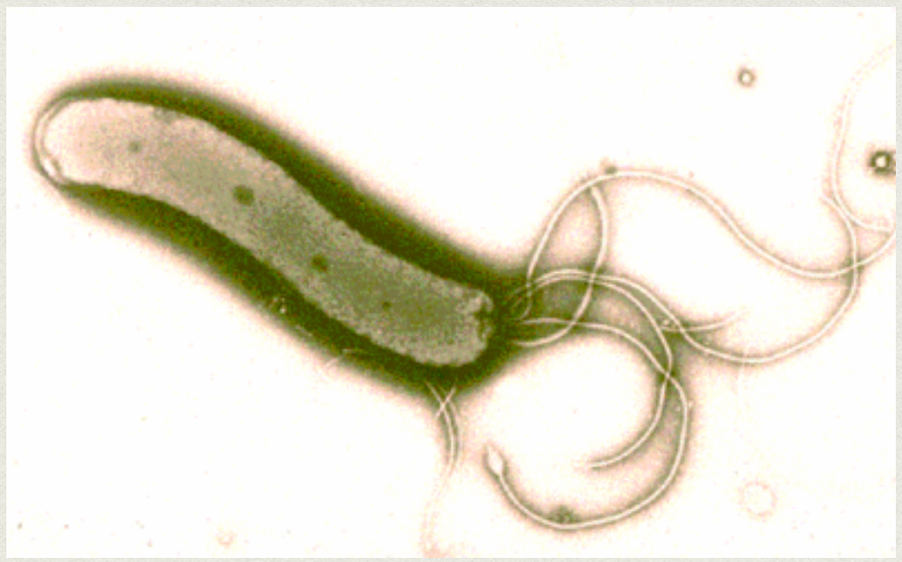
RecA DNA sequence from *Helicobacter pylori* and *Streptococcus mutant*, window=1 match=1

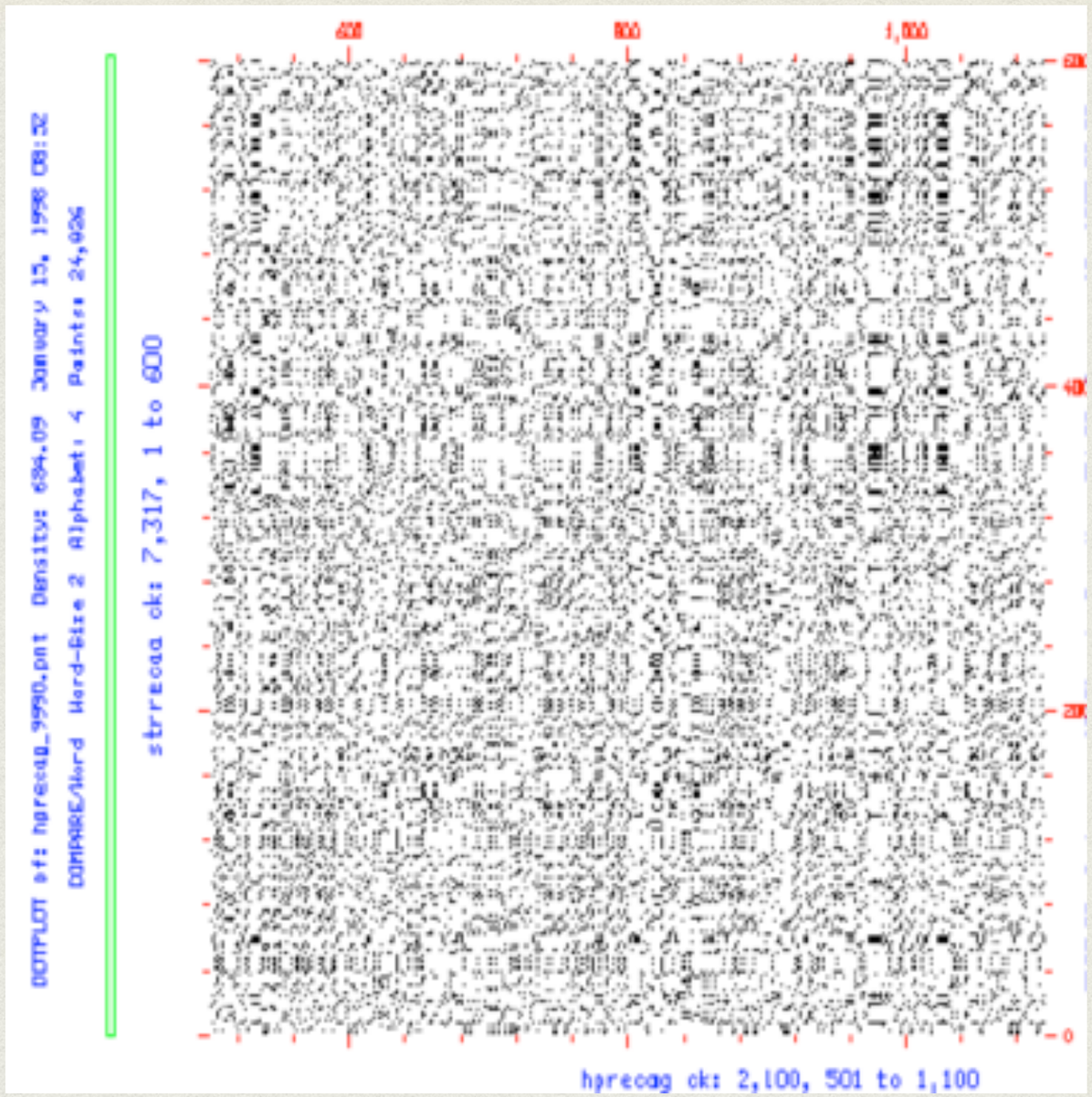




DOT PLOT - EXAMPLES

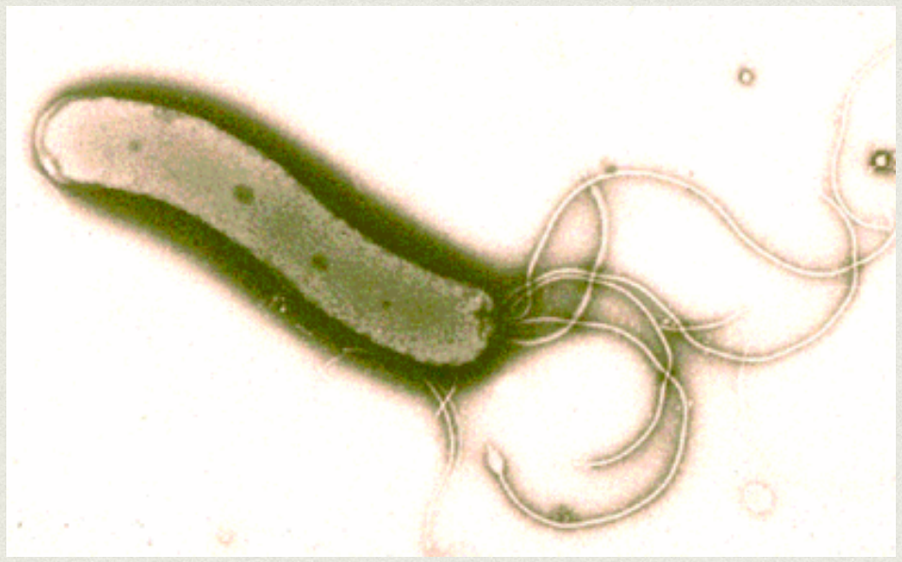
RecA DNA sequence from *Helicobacter pylori* and *Streptococcus mutant*, window=2 match=2

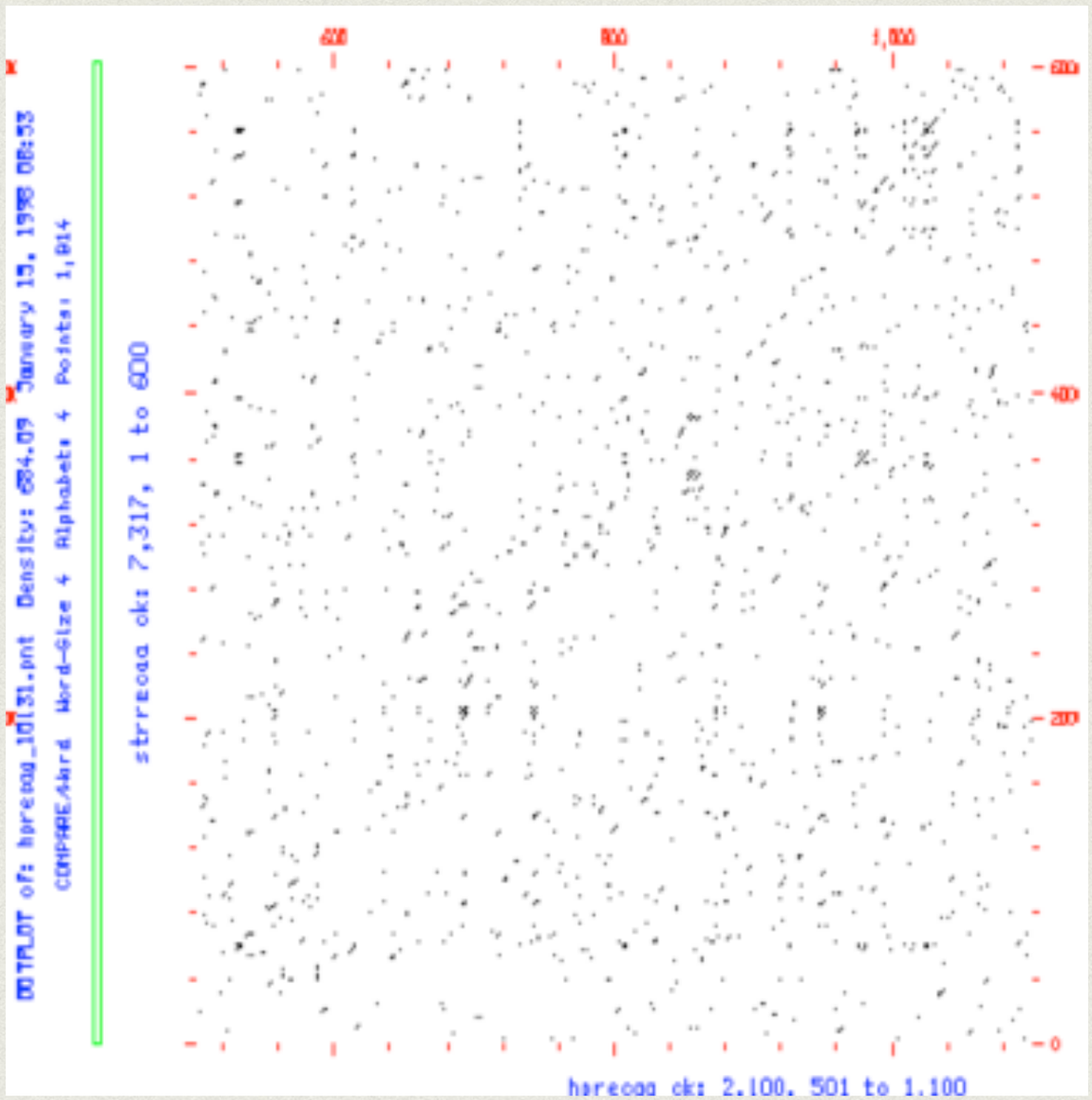




DOT PLOT - EXAMPLES

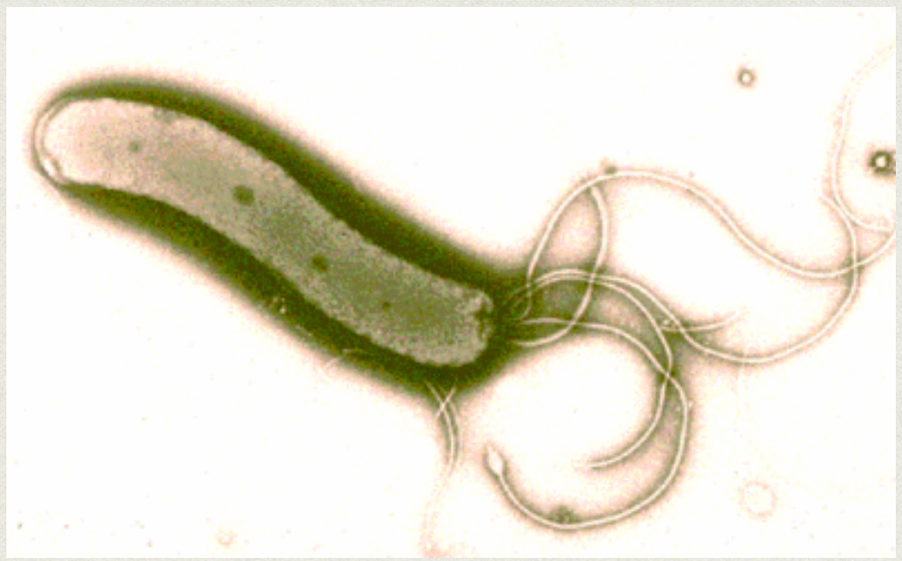
RecA DNA sequence from *Helicobacter pylori* and *Streptococcus mutant*, window=4 match=4

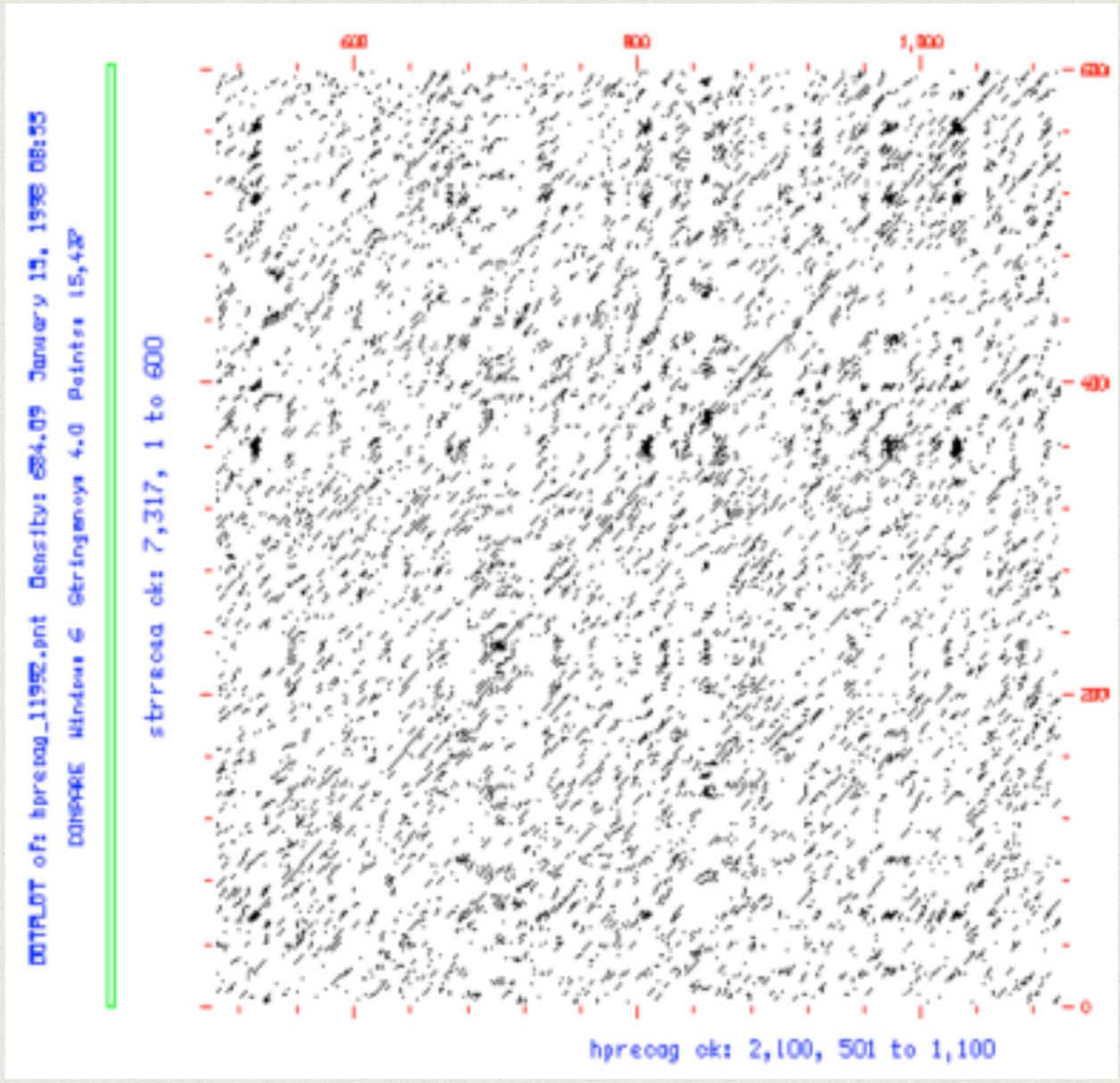




DOT PLOT - EXAMPLES

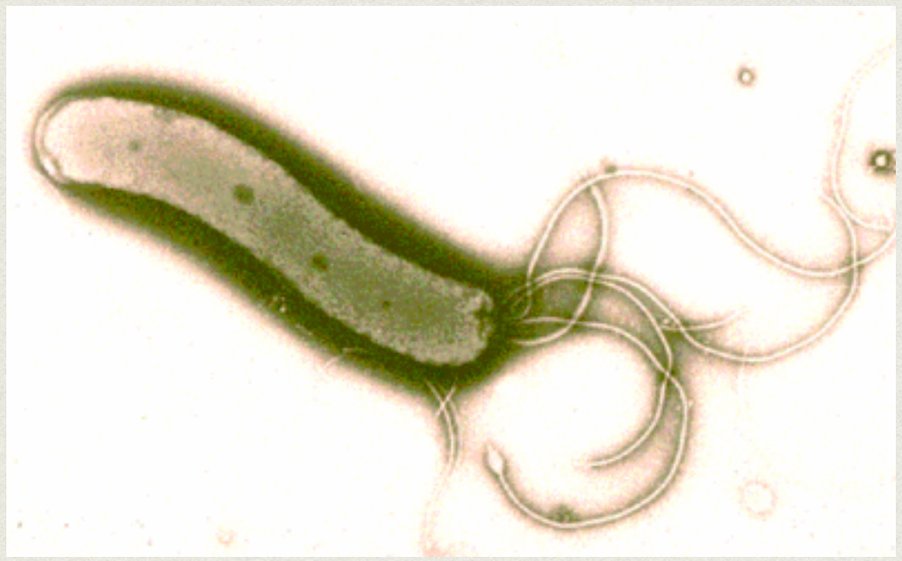
RecA DNA sequence from *Helicobacter pylori* and *Streptococcus mutant*, window=6 match=4

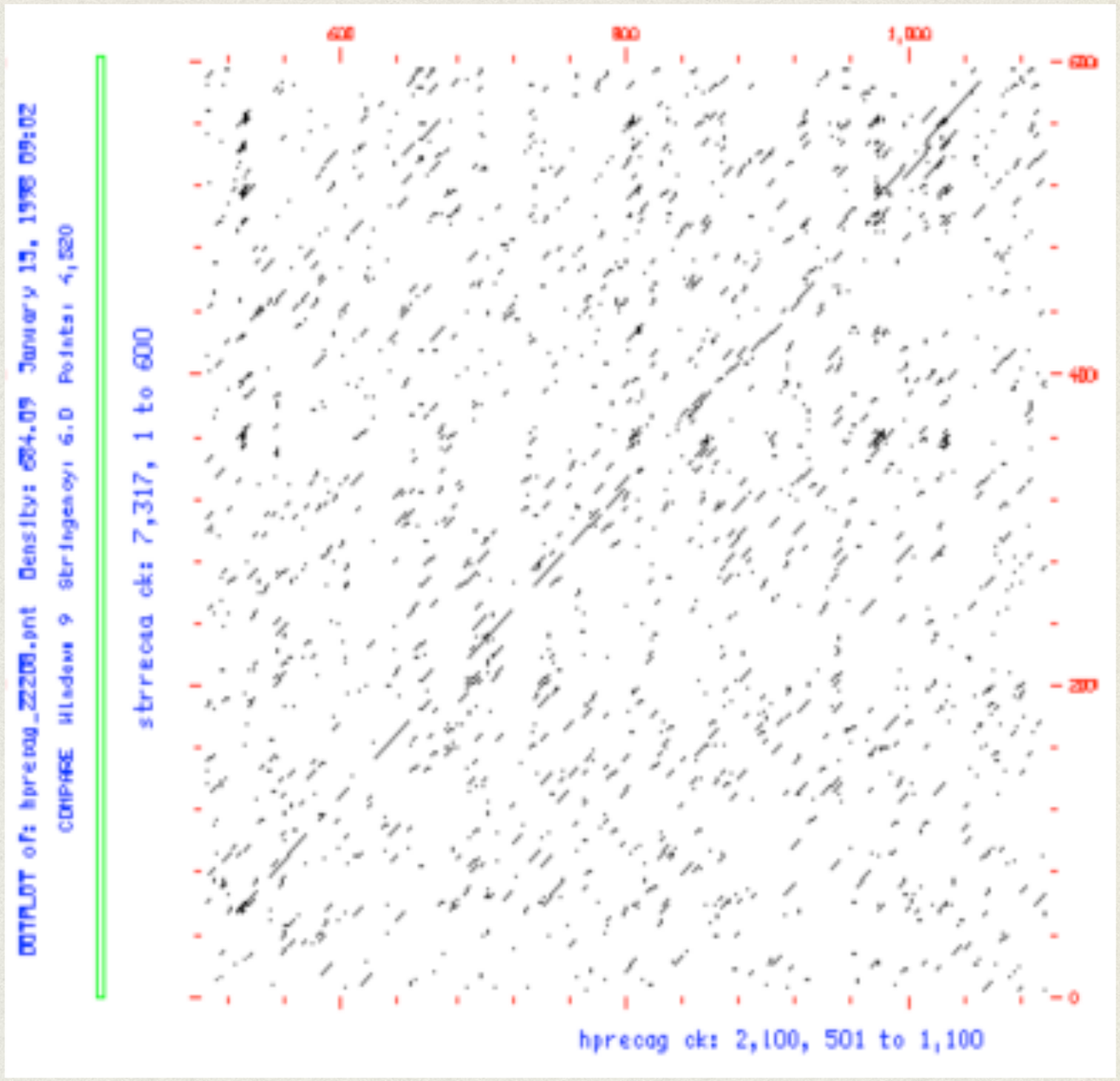




DOT PLOT - EXAMPLES

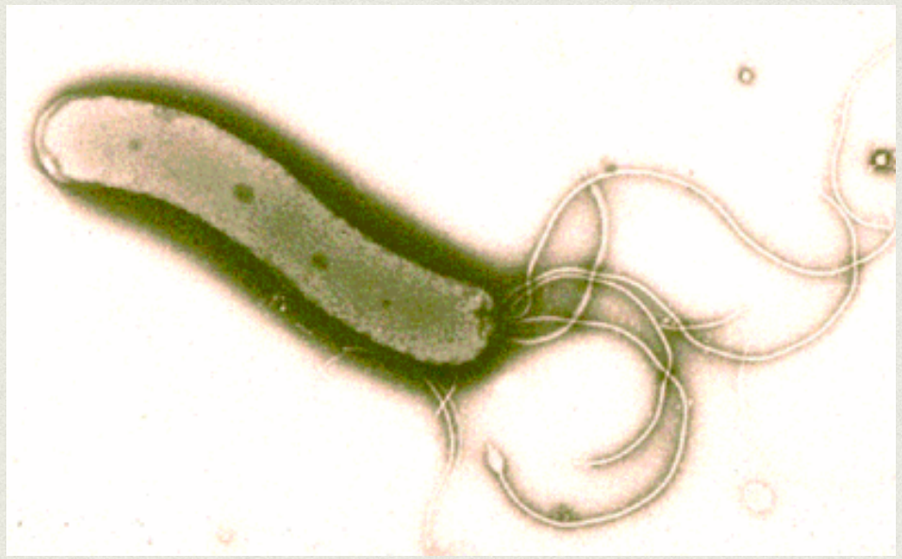
RecA DNA sequence from *Helicobacter pylori* and Streptococcus *mutant*, window=9 match=6

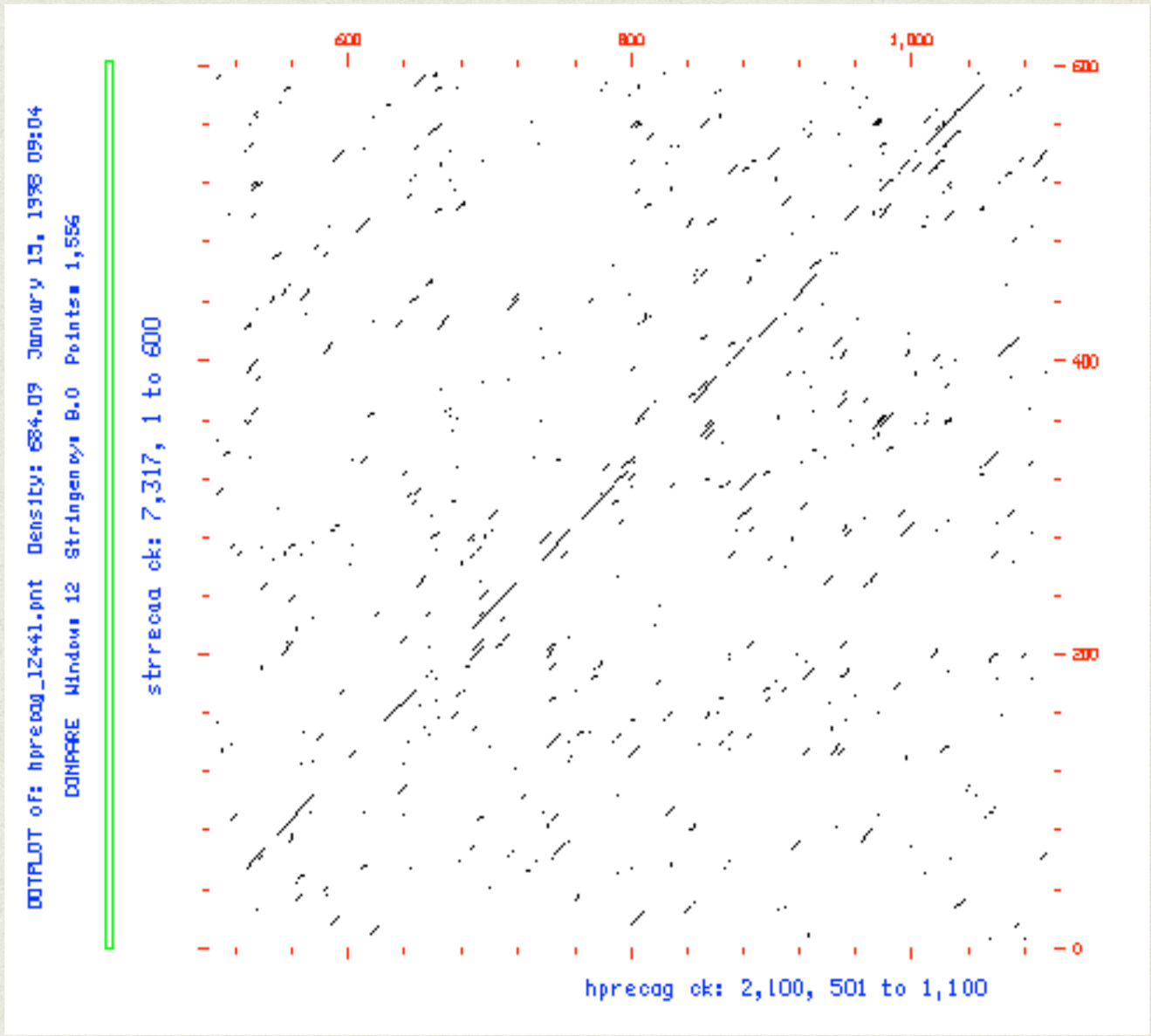




DOT PLOT - EXAMPLES

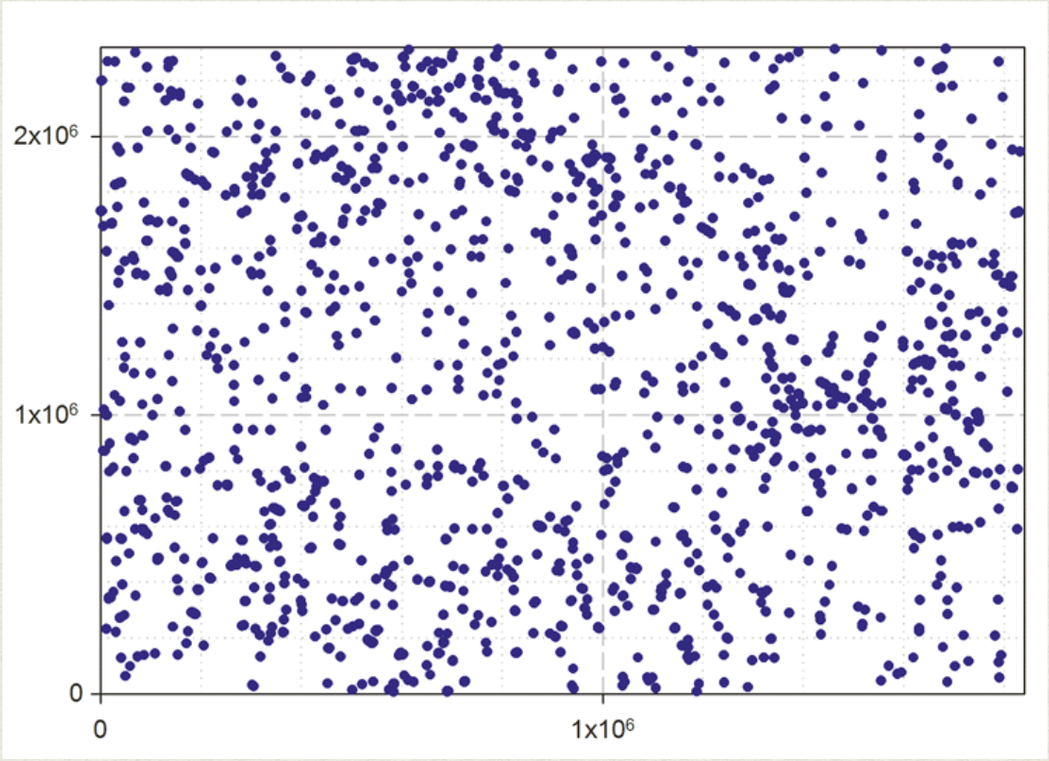
RecA DNA sequence from *Helicobacter pylori* and *Streptococcus mutant*, window=12 match=8





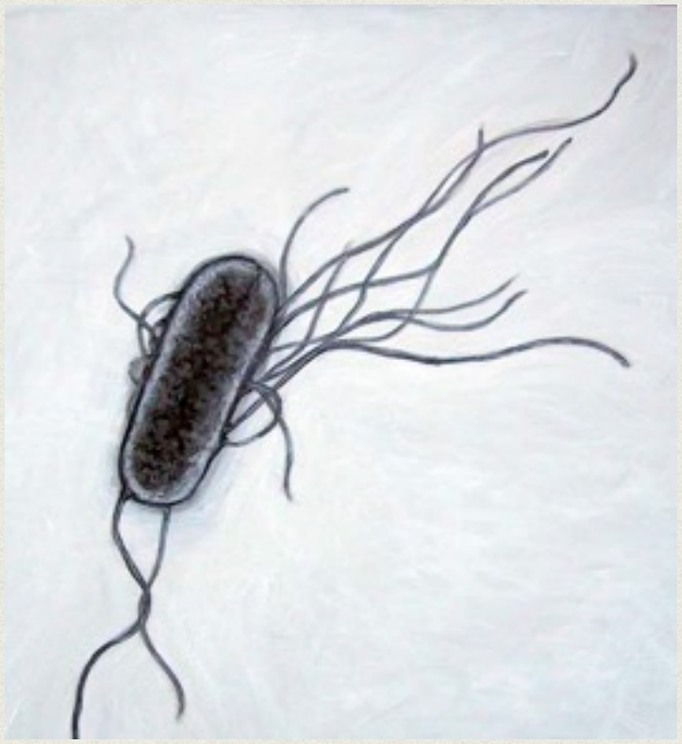
DOT PLOT - WHAT CAN YOU SEE THERE?

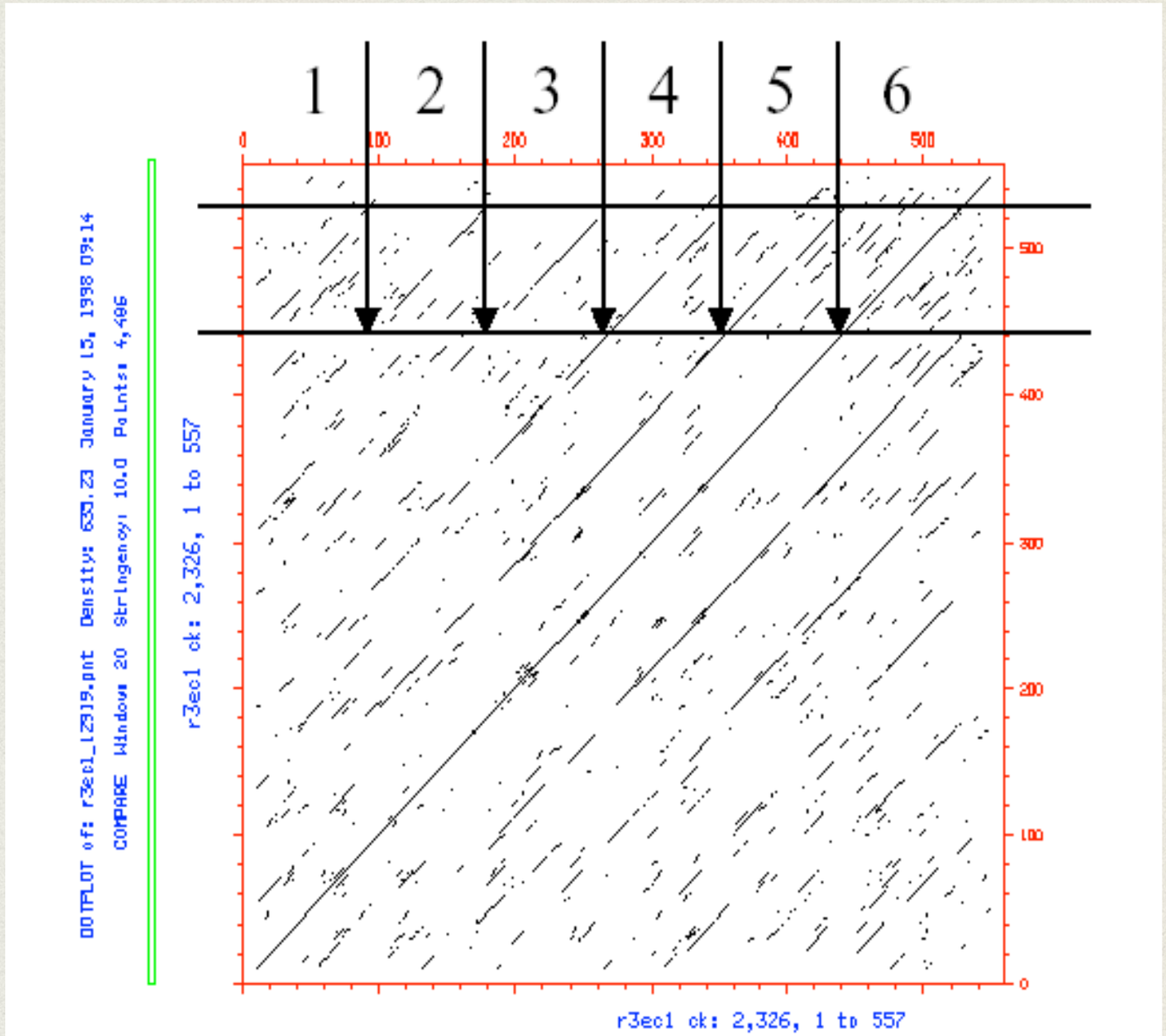
- Similar regions
- Repeated sequences
- Sequence rearrangements
- RNA structures
- Gene order



DOT PLOT EXAMPLES - REPEATS

Repeated sequence in *Escherichia coli* ribosomal protein S1



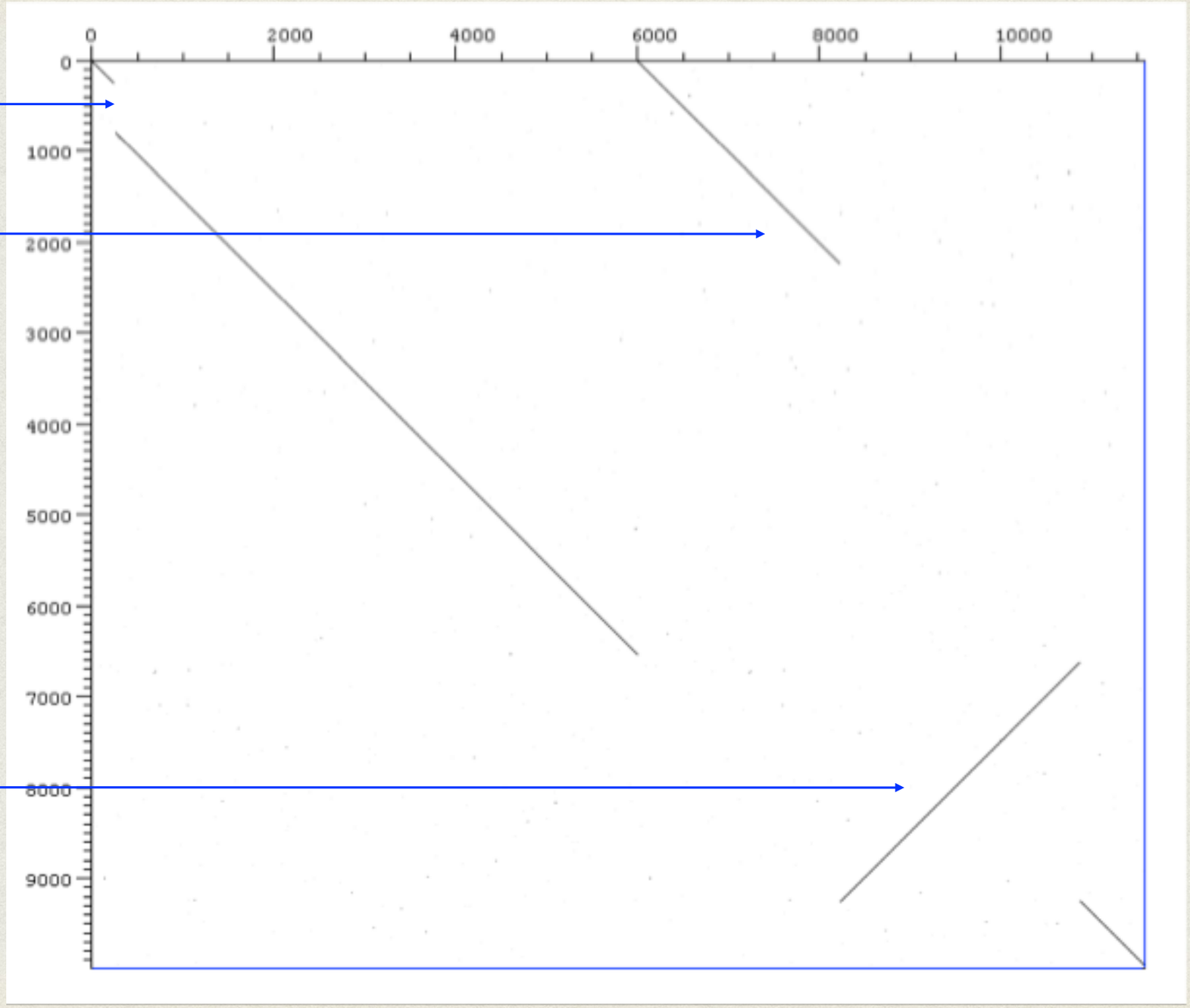


DOT PLOT EXAMPLES - REARRANGEMENTS

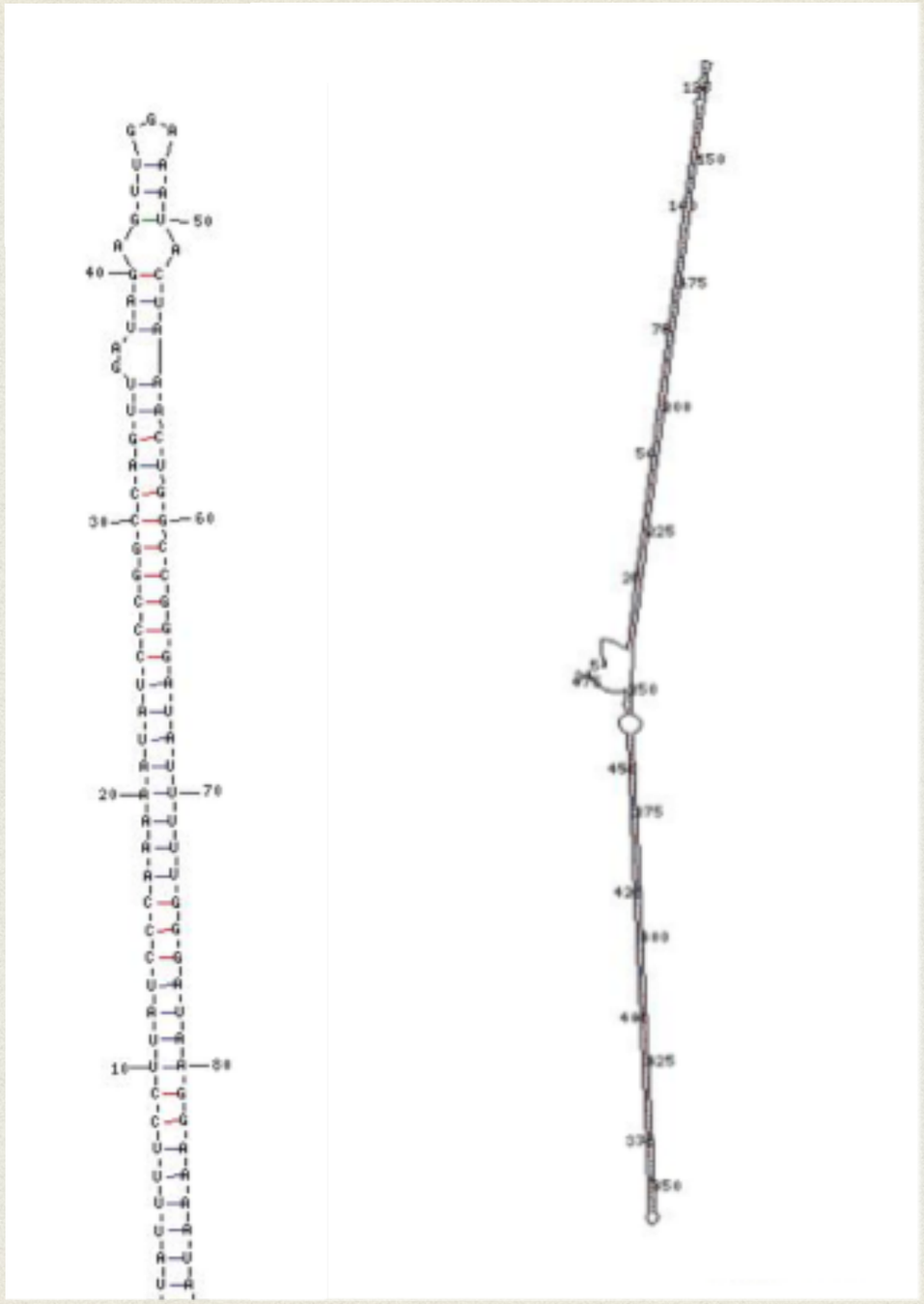
deletion

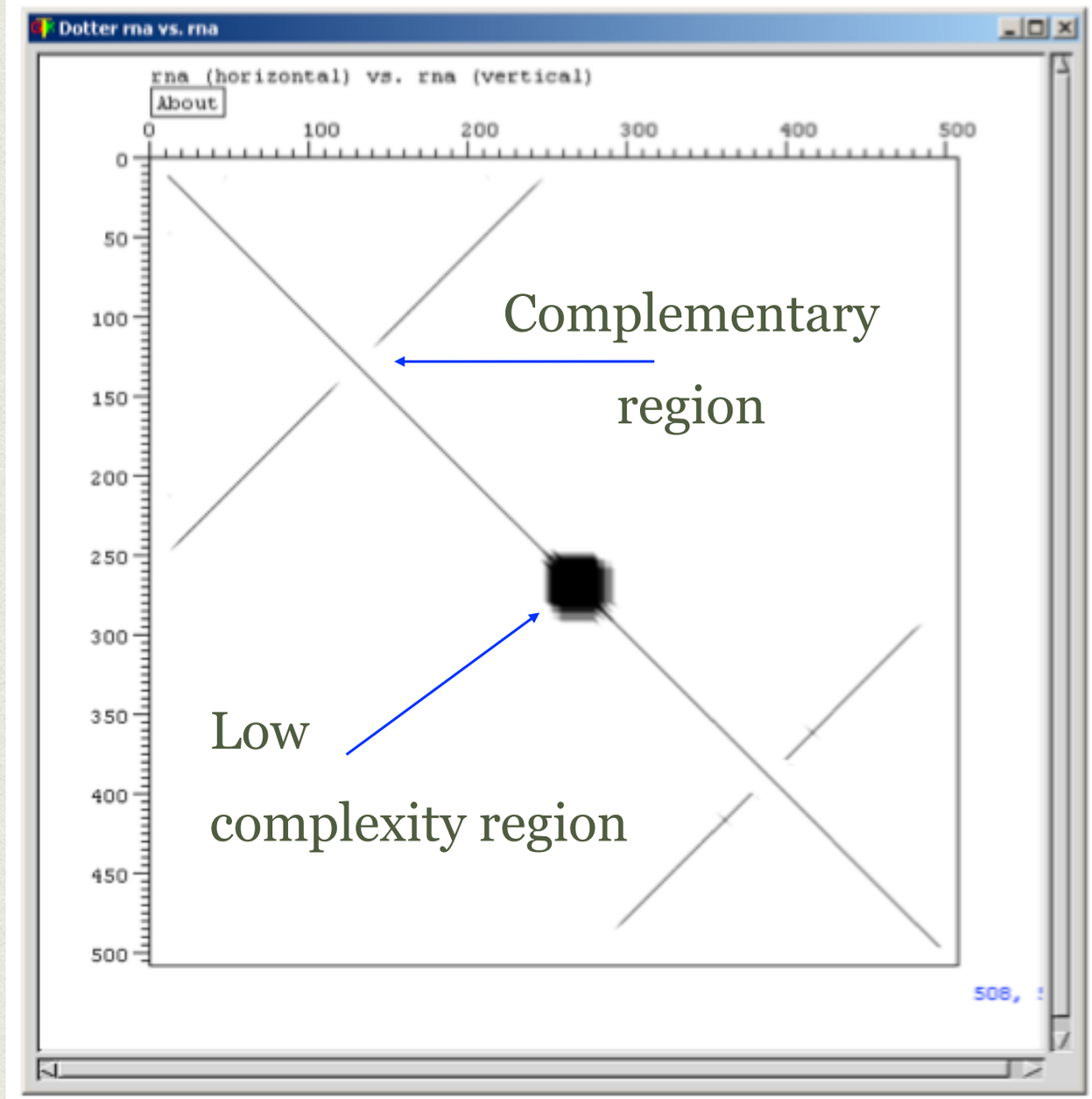
duplication

inversion



DOT PLOT EXAMPLES - RNA STRUCTURE





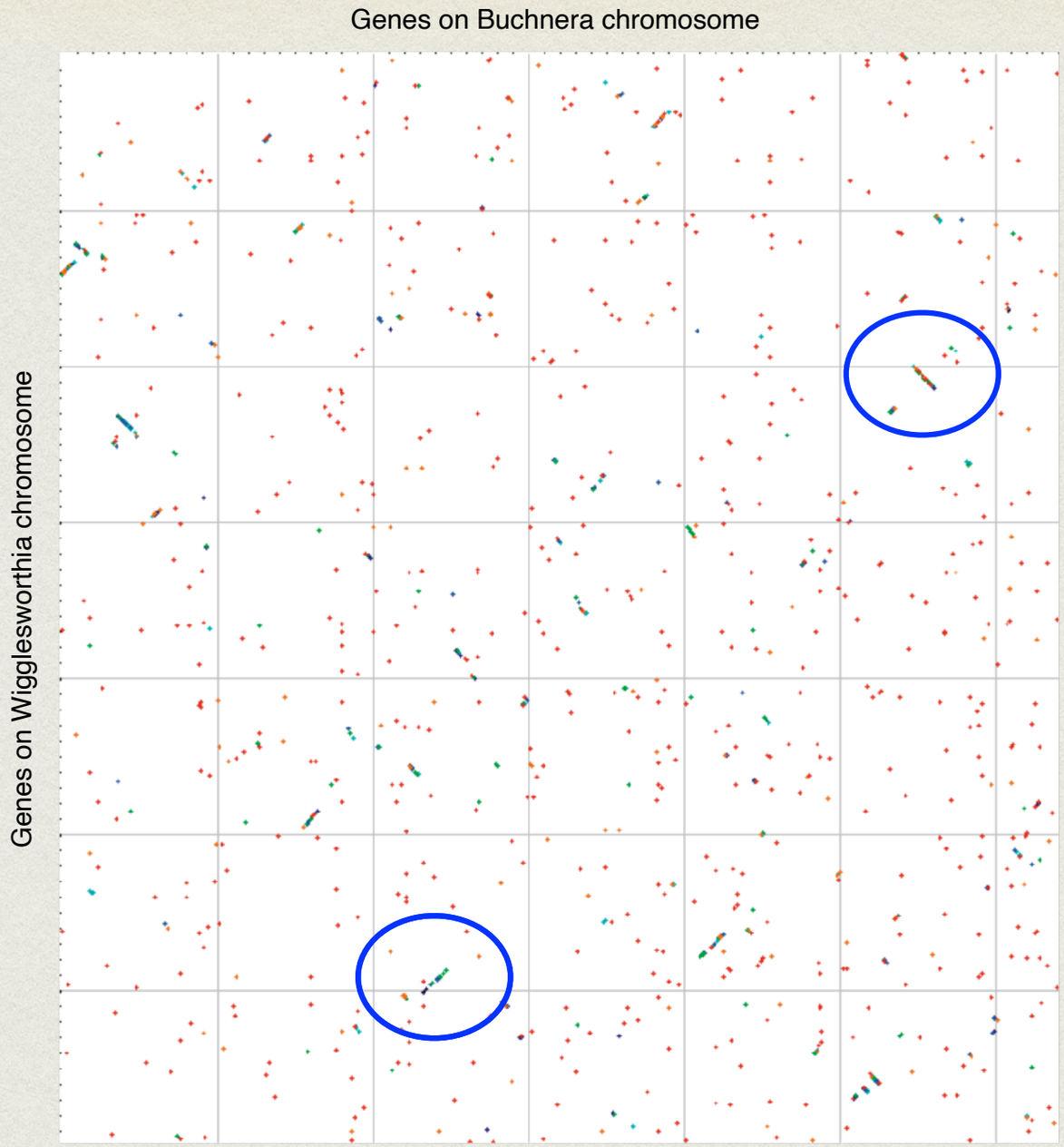
DOT PLOT EXAMPLES - GENE ORDER

Whole genome comparison of *Buchnera* against *Wigglesworthia*

Each dot represents genes that are similar between two genomes as defined by BLAST search

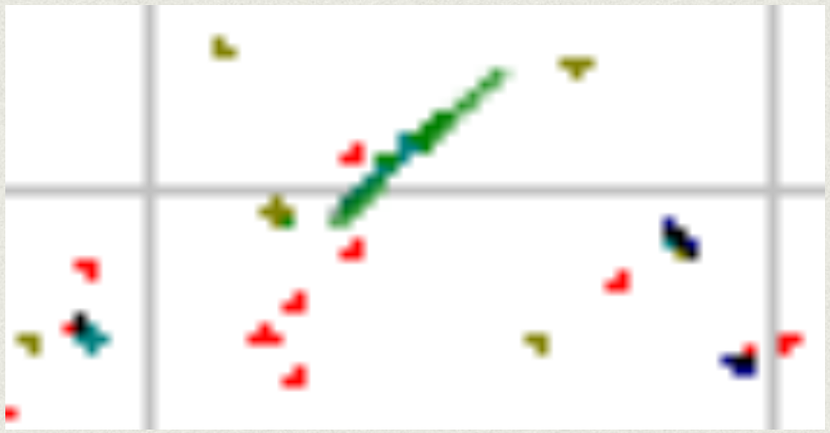
red dots - genes on the same strand

green dots - genes on opposite strand



DOT PLOT EXAMPLES - POTENTIAL OPERONS

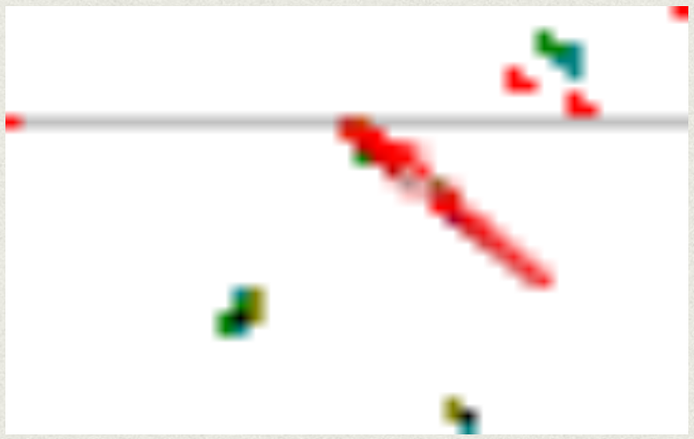
Whole genome comparison of *Buchnera* against *Wigglesworthia*



Each dot represents genes that are similar between two genomes as defined by BLAST search

red dots - genes on the same strand

green dots - genes on opposite strand



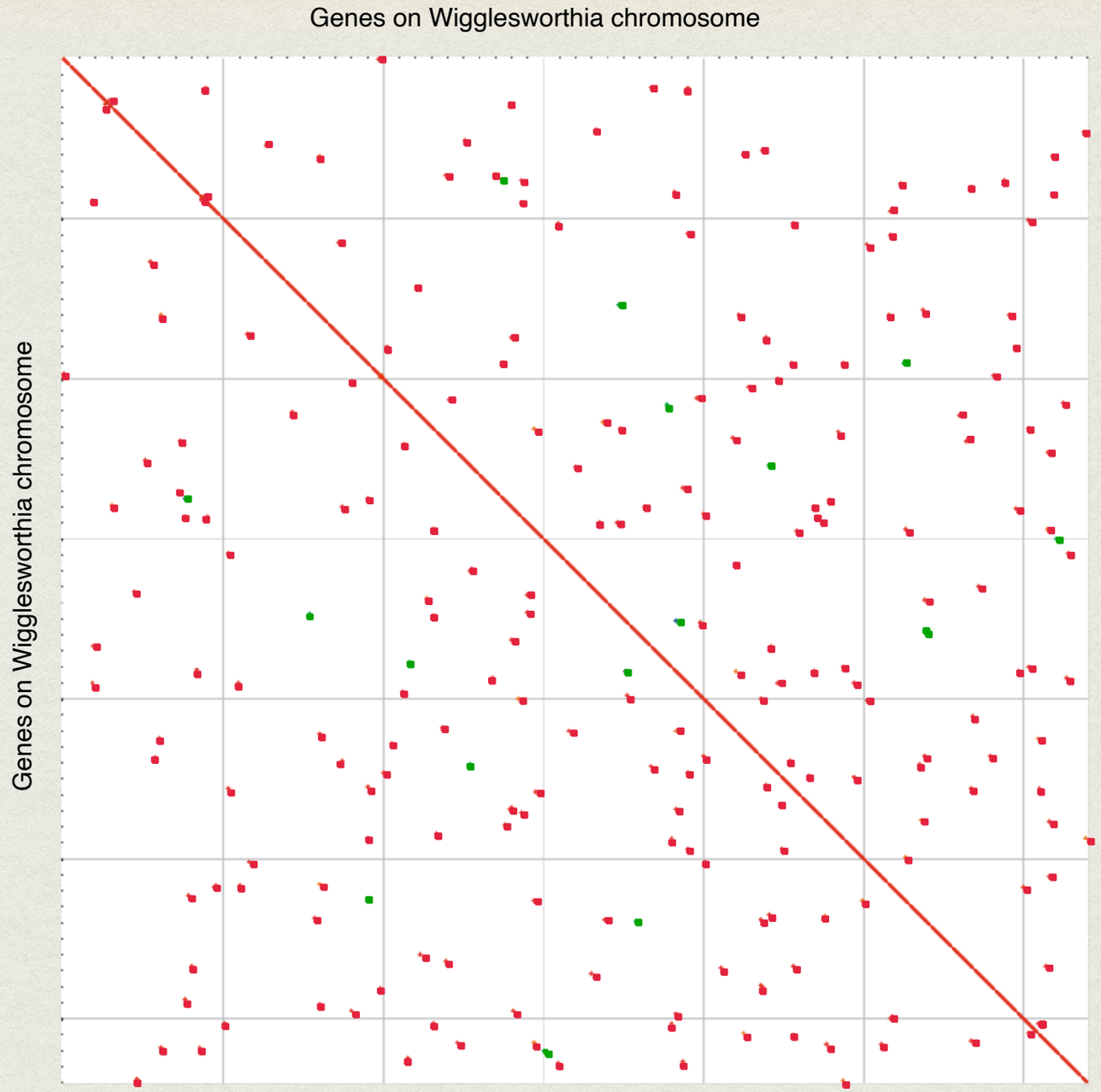
DOT PLOT EXAMPLES - PARALOGOUS GENES

Whole genome comparison of *Wigglesworthia*

red dots - paralogs on the same strand

green dots - paralogs on opposite strand

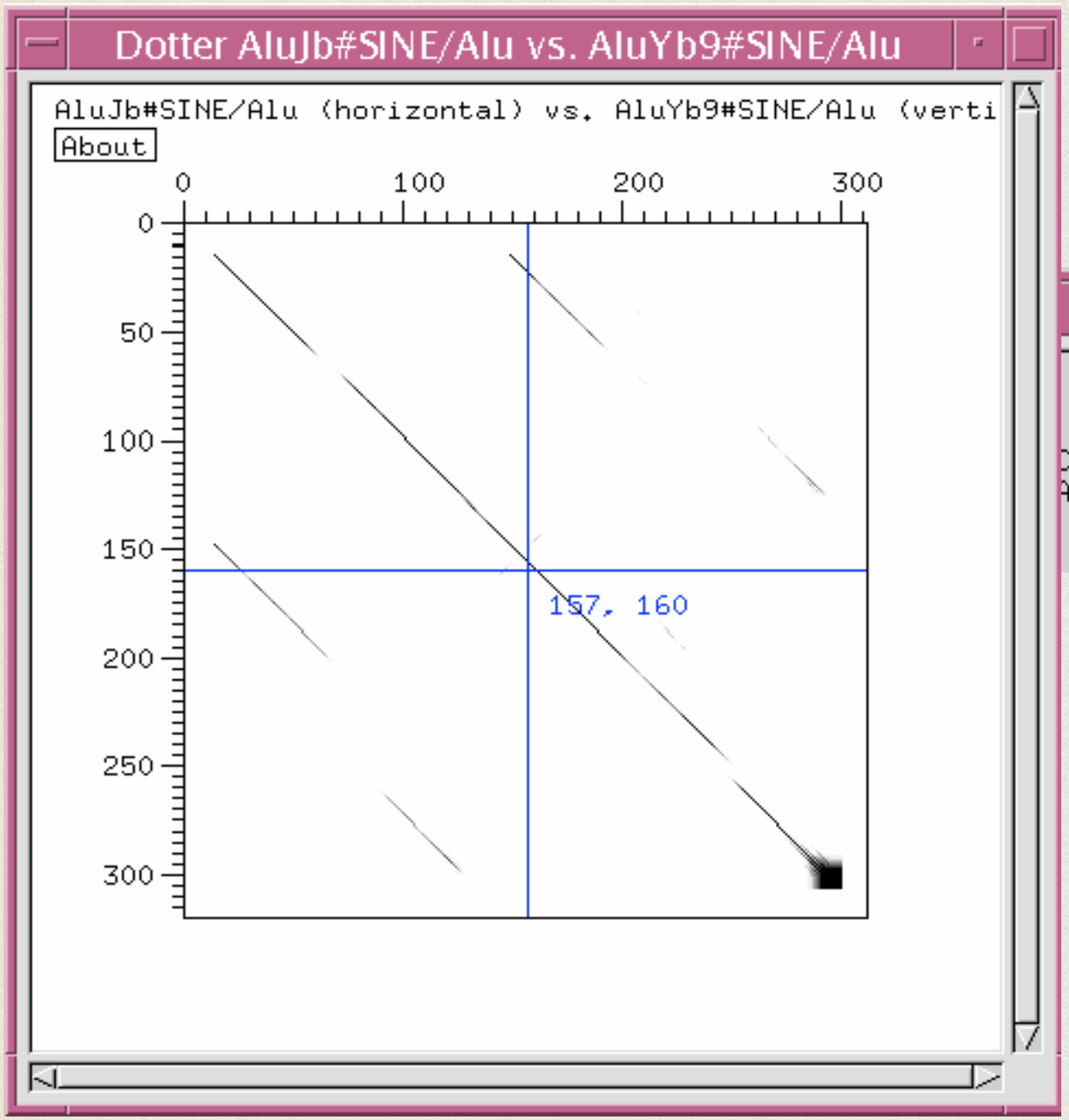
Note: self-hits of all genes form red diagonal line

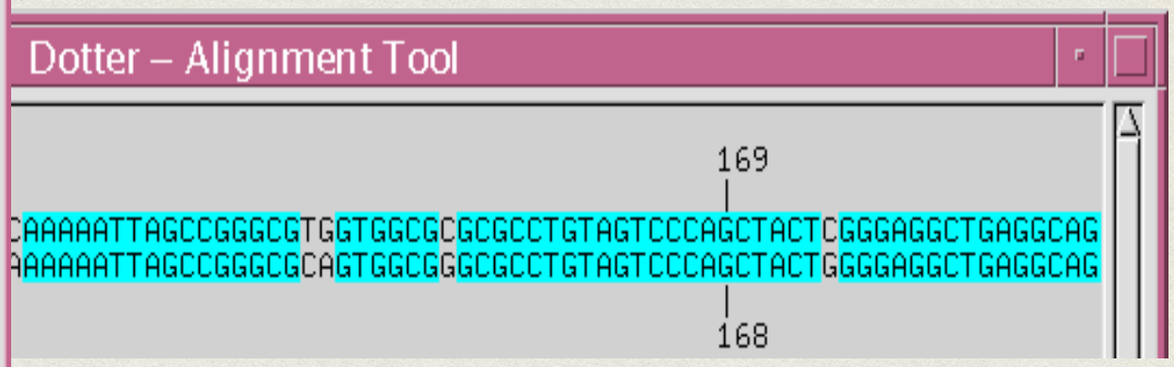


DOT PLOTS RULES OF THUMB

- Don't get too many points, about 3-5 times the length of the sequence is about right (1-2%)
- Window size about 20 for distant proteins and12 for nucleic acid (try stringency 50%)
- Check sequence against itself
  - Finds internal repeats
- Check sequence against another sequence
  - Finds repeats and rearrangements
- The best programs should have dynamic adjustment of parameters
  - dotlet: https://dotlet.vital-it.ch
  - gepard: http://cube.univie.ac.at/gepard

DOT PLOTS VERSUS ALIGNMENTS





ALIGNMENT

- Linear representation of relation between sequences that shows one-to-one correspondence between amino acid or nucleotide residue

- How can we define a quantitative measure of sequence similarity?
  - match
  - mismatch
  - gap

```
gctg-aa-cg
-ctataa-tc
```

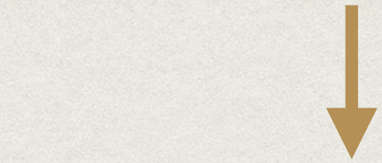
ALIGNMENT PROBLEM

THISISCOMPLETELYNEWSEQUENCE

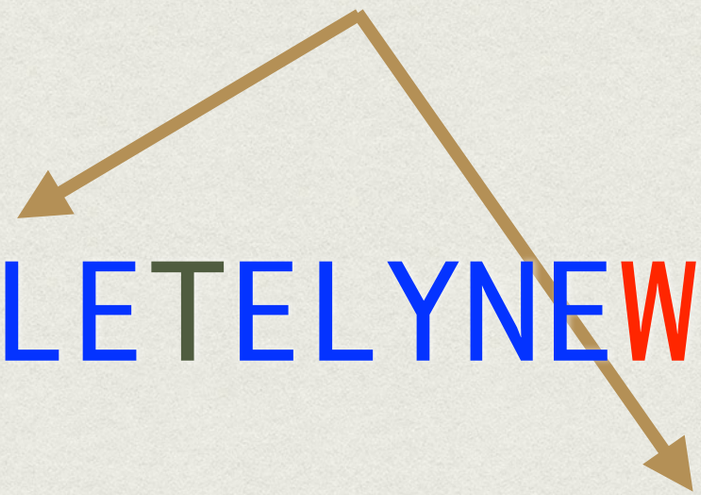
THISISSUPEREXTRASEQUENCE

EVOLUTIONARY BASIS OF SEQUENCE ANALYSES

THISISANANCESTRALSEQUENCE



THISISCNMPESTRAWSEQUENCE



THISISCOMPLETELYNEWSEQUENCE

THISISSUPEREXTRASEQUENCE

ALIGNMENT PROBLEM

THISISANANCESTRALSEQUENCE
THISISCOMPLETELYNEWSEQUENCE

THISISANANCESTRALSEQUENCE
THISISSUPEREXTRASEQUENCE

ALIGNMENT PROBLEM

THISISANANCEST-R--ALSEQUENCE
THISISCOMP-LETELYNEWSEQUENCE

THISISANANCES-TRALSEQUENCE
THISISSU-PEREXTRA-SEQUENCE

ALIGNMENT PROBLEM

THISISCOMP-LETELYNEWSEQUENCE
THISISANANCEST-R--ALSEQUENCE

THISISANANCES-TRALSEQUENCE
THISISSU-PEREXTRA-SEQUENCE

ALIGNMENT PROBLEM

```
THISISCOMP-LE-TELYNEWSEQUENCE
THISISANANCES-T-R--ALSEQUENCE
THISISANANCES-T-R--ALSEQUENCE
THISISSU-PEREXT-R--A-SEQUENCE
```

ALIGNMENT PROBLEM

```
THISISCOMP-LE-TELYNEWSEQUENCE
THISISSU-PEREXT-R--A-SEQUENCE
```

The problem is that we need to model evolutionary events based on extant sequences, without knowing an ancestral one!

CLASSIFICATION OF SEQUENCE ALIGNMENTS

Global alignment

```
THISISCOMP-LE-TELYNEWSEQUENCE
THISISSU-PEREXT-R--A-SEQUENCE
```

Local alignment

```
THISISCOMPLETELYNEWSEQUENCE
     COMP-ETE
```

GLOBAL VS. LOCAL ALIGNMENT

- Global alignment algorithms start at the beginning of two sequences and add gaps to each sequence until the end of one of the sequences is reached.

- Local alignment algorithms find the region(s) of highest similarity between two sequences and build the alignment outward from there.

CLASSIFICATION OF SEQUENCE ALIGNMENTS

Pairwise alignment

```
THISISCOMP-LE-TELYNEWSEQUENCE
THISISSU-PEREXT-R--A-SEQUENCE
```

Multiple sequence alignment

```
THISISCOMP-LE-TELYNEWSEQUENCE
THISISANANCES-T-R--ALSEQUENCE
THISISSU-PEREXT-R--A-SEQUENCE
```

ALIGNMENT

- Any assignment of correspondences that preserves the order of residues within the sequence is an alignment
- It is the basic tool of bioinformatics
- Computational challenge - introduction of insertions and deletions (gaps) that correspond to evolutionary events
- We must define criteria so that an algorithm can choose the best alignment

ALIGNMENT
AN EXAMPLE

Let's compare two strings `gctgaacg` and `ctataatc`

an uninformative alignment

```
-------gctgaacg
ctataatc-------
```

an alignment without gaps

```
gctgaacg
ctataatc
```

an alignment with gaps

```
gctga-a--cg
--ct-ataatc
```

another alignment with gaps

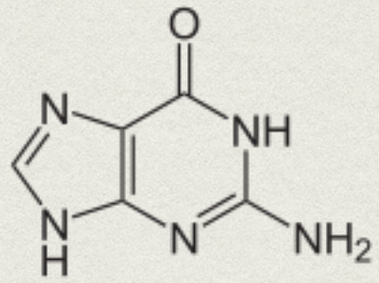
```
gctg-aa-cg
-ctataa-tc
```

SCORING SCHEMES

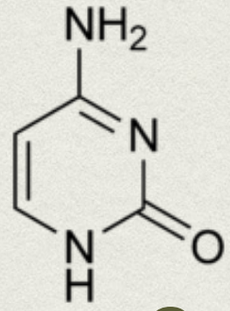
- A scoring system must account for residue substitution, and insertions or deletions (indels)
- Indels (gaps) will have scores that depend on their length
- For nucleic acid sequences, it is common to use a simple scheme for substitutions, e.g. +1 for a match, -1 for a mismatch
- More realistic would be to take into account nucleotide frequencies (sequence composition) and fact that transitions are more frequent than transversions

Purines

Pyrimidines



Transversions



Guanine

Cytosine

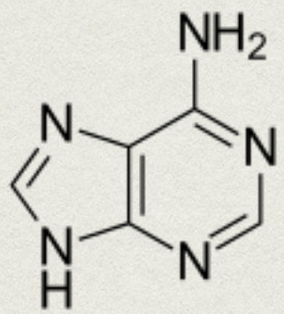
Transitions

Transitions

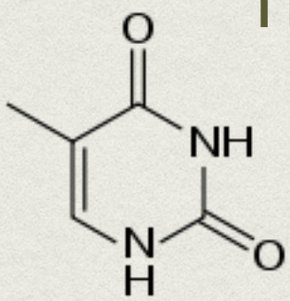
Transversions

Transversions

Adenine



Transversions



Thymine

SCORING SCHEMES

- A scoring system must account for residue substitution, and insertions or deletions (indels)
- Indels (gaps) will have scores that depend on their length
- For nucleic acid sequences, it is common to use a simple scheme for substitutions, e.g. +1 for a match, -1 for a mismatch
- More realistic would be to take into account nucleotide frequencies (sequence composition) and fact that transitions are more frequent than transversions
- LAST (http://last.cbrc.jp) software is using *ad hoc* built scoring matrix based on sequences to be aligned

GAP SCORING SYSTEMS

- non-affine model - each gap position treated the same, e.g. match = 4, mismatch = -3, gap -4

- affine model - first gap position penalized more than others, e.g. match = 4, mismatch = -3, gap opening = -8, gap = -4

GAP SCORING AN EXAMPLE

non-affine gapping score - the second alignment is "better"

```
GGTGCCAC-TCCAC------CTG
AGTGCCACCCCCCAATGCCGCTG
```

-3 4 4 4 4 4 4 4 -4 -3 4 4 4 -3 -4 -4 -4 -4 -4 4 4 4 = 23

```
GGTGCCAC-TCCA---C--CTG
AGTGCCACCCCCCAATGCCGCTG
```

-3 4 4 4 4 4 4 4 -4 -3 4 4 4 -4 -4 -4 4 -4 -4 4 4 4 = 26

GAP SCORING AN EXAMPLE

affine gapping score - the first alignment is "better"

```
GGTGCCAC-TCCAC------CTG
AGTGCCACCCCCCAATGCCGCTG
```

-3 4 4 4 4 4 4 4 **-12** -3 4 4 4 -3 **-12** -4 -4 -4 -4 4 4 4 = 7

```
GGTGCCAC-TCCA---C--CTG
AGTGCCACCCCCCAATGCCGCTG
```

-3 4 4 4 4 4 4 4 **-12** -3 4 4 4 **-12** -4 -4 4 **-12** -4 4 4 4 = 2

GAP SCORING AN EXAMPLE

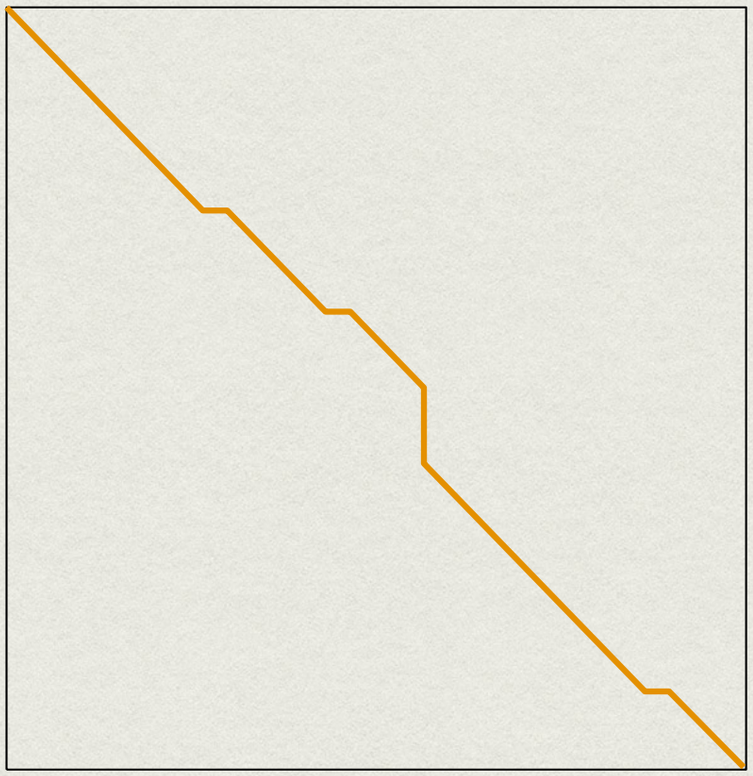
Equivalent alignments

```
GGTGCCAC-TCCA---C--CTG
AGTGCCACCCCCCAATGCCGCTG
-3  4  4  4  4  4  4  4 -12 -3  4  4  4 -12 -4 -4  4 -12 -4  4  4  4  =  2
```

```
GGTGCCACT-CCA---C--CTG
AGTGCCACCCCCCAATGCCGCTG
-3  4  4  4  4  4  4  4 -3 -12 4  4  4 -12 -4 -4  4 -12 -4  4  4  4  =  2
```

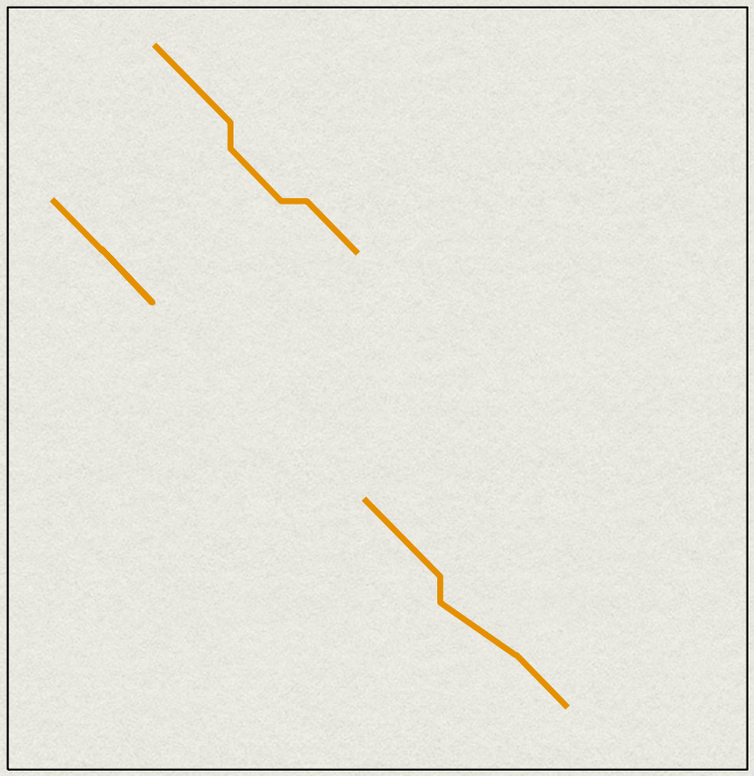
GLOBAL VERSUS LOCAL ALIGNMENT

Optimal global alignment



Sequences align essentially from end to end.

Optimal local alignment



Sequences align only in small, isolated regions.

SEQUENCE ALIGNMENT

- Brute-force approach
  - generate all possible alignments between two sequences and score them
  - the alignment(s) with the best score is an optimal one
- Problem
  - computationally too expensive
    - there are about $2^{2N}/\sqrt{(2\pi N)}$ different alignments for two equal sequences of length N
    - for two sequences of length 300, that's about 10^{179} different alignments

Sequence alignment using dynamic programming

DYNAMIC PROGRAMMING

- dynamic programming (also known as dynamic optimization) is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions

- it avoids computing the same results over and over again

DYNAMIC PROGRAMMING

- Three steps:

(1) break the problem into smaller sub-problems

(2) solve the smaller problems optimally

(3) use the sub-problem solutions to construct and optimal solution for the original problem

THE NEEDLEMAN-WUNSCH ALGORITHM

$$S(i,j) = \max \begin{cases} S(i-1, j-1) + \sigma(x_i, y_j) \\ S(i-1, j) + \gamma \\ S(i, j-1) + \gamma \end{cases}$$

where σ - match/mismatch score and γ - gap penalty

Needleman, Saul B. & Wunsch, Christian D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*. 48 (3): 443–53.

DYNAMIC PROGRAMMING

Construct an optimal alignment of these two sequences:

```
G A T A C T A
G A T T A C C A
```

Using these scoring rules:

Match:	+1
Mismatch:	-1
Gap:	-1

DYNAMIC PROGRAMMING

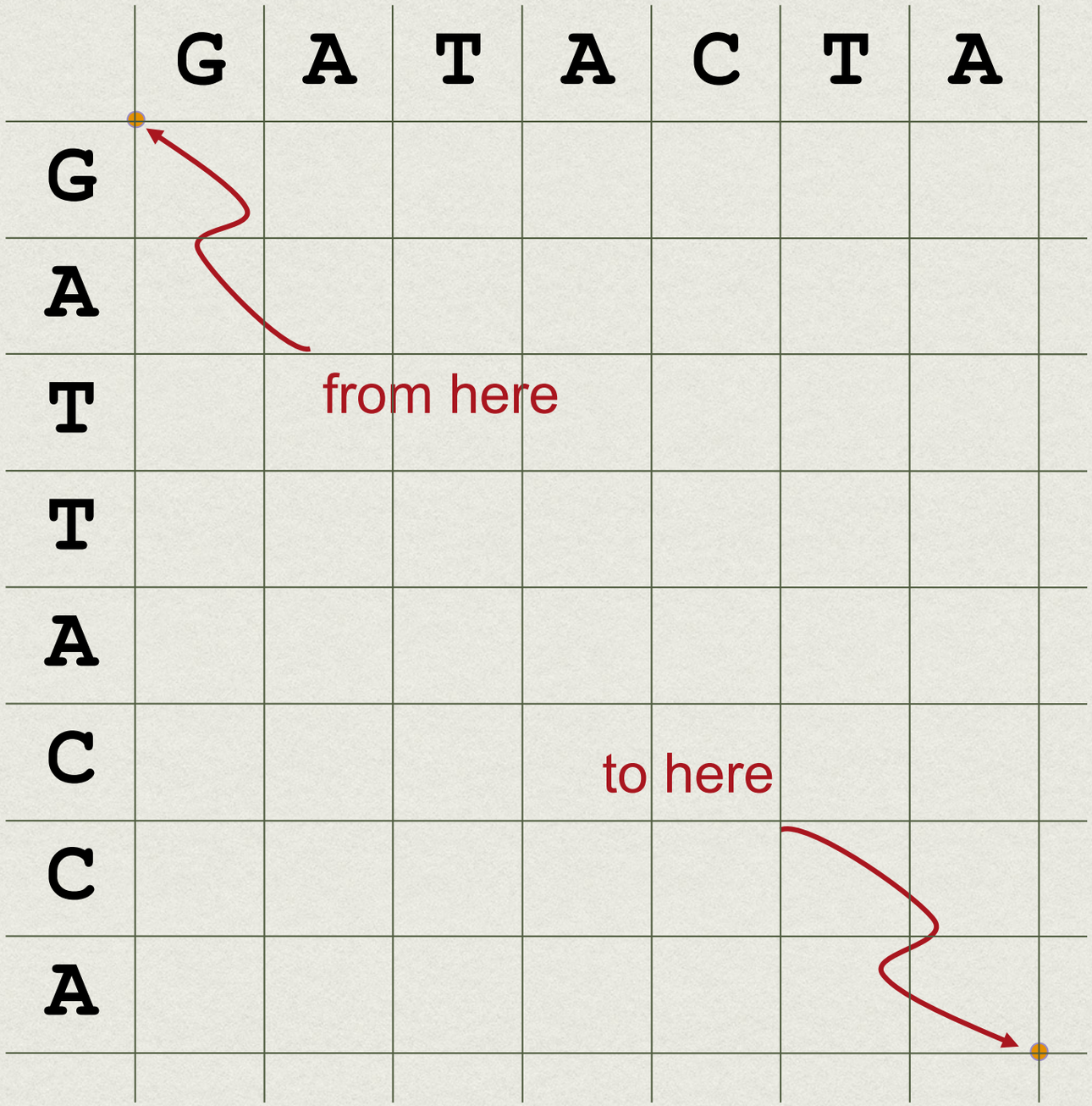
Arrange the sequence residues along a two-dimensional lattice

Vertices of the lattice fall between letters

	G	A	T	A	C	T	A	
G								
A								
T								
T								
A								
C								
C								
A								

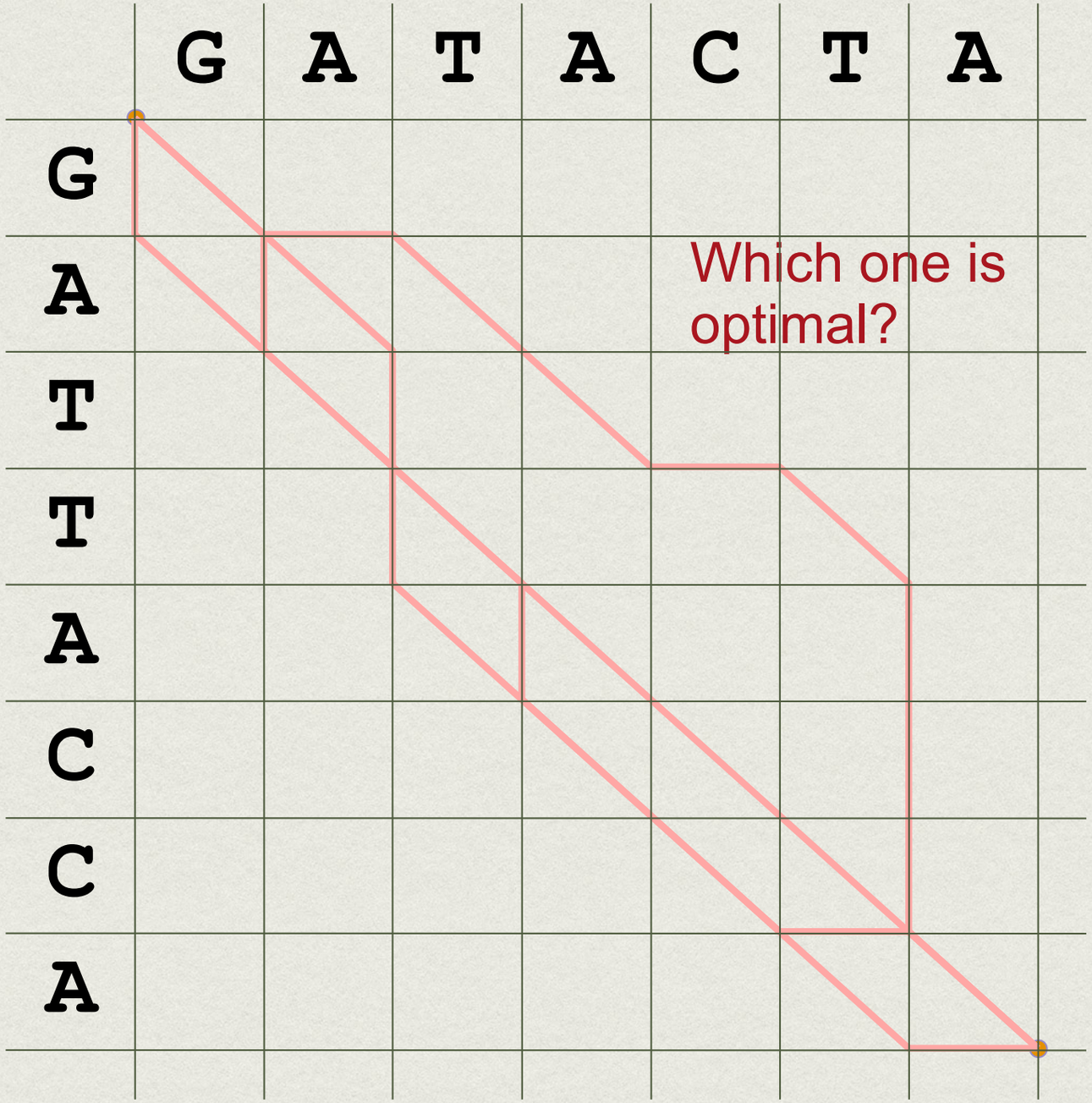
DYNAMIC PROGRAMMING

The goal is to find the optimal path



DYNAMIC PROGRAMMING

Each path corresponds to a unique alignment



DYNAMIC PROGRAMMING

The score for a path is the sum of its incremental edges scores

		G	A	T	A	C	T	A	
	G			**A** aligned with **A**					
	A			Match = +1					
	T								
	T								
	A								
	C								
	C								
	A								

Match: +1

Mismatch: -1

Gap: -1

DYNAMIC PROGRAMMING

The score for a path is the sum of its incremental edges scores

		G	A	T	A	C	T	A	
	G								
	A				**A** aligned with **T**				
	T				Mismatch = -1				
	T								
	A								
	C								
	C								
	A								

Match: +1

Mismatch: -1

Gap: -1

Dynamic programming

The score for a path is the sum of its incremental edges scores

	G	A	T	A	C	T	A	
G								
A								
T								
T								
A								
C								
C								
A								

T aligned with *NULL*

Gap = -1

NULL aligned with **T**

Match: +1

Mismatch: -1

Gap: -1

Dynamic programming

Incrementally extend the path

		G	A	T	A	C	T	A	
	0								
G									
A									
T									
T									
A									
C									
C									
A									

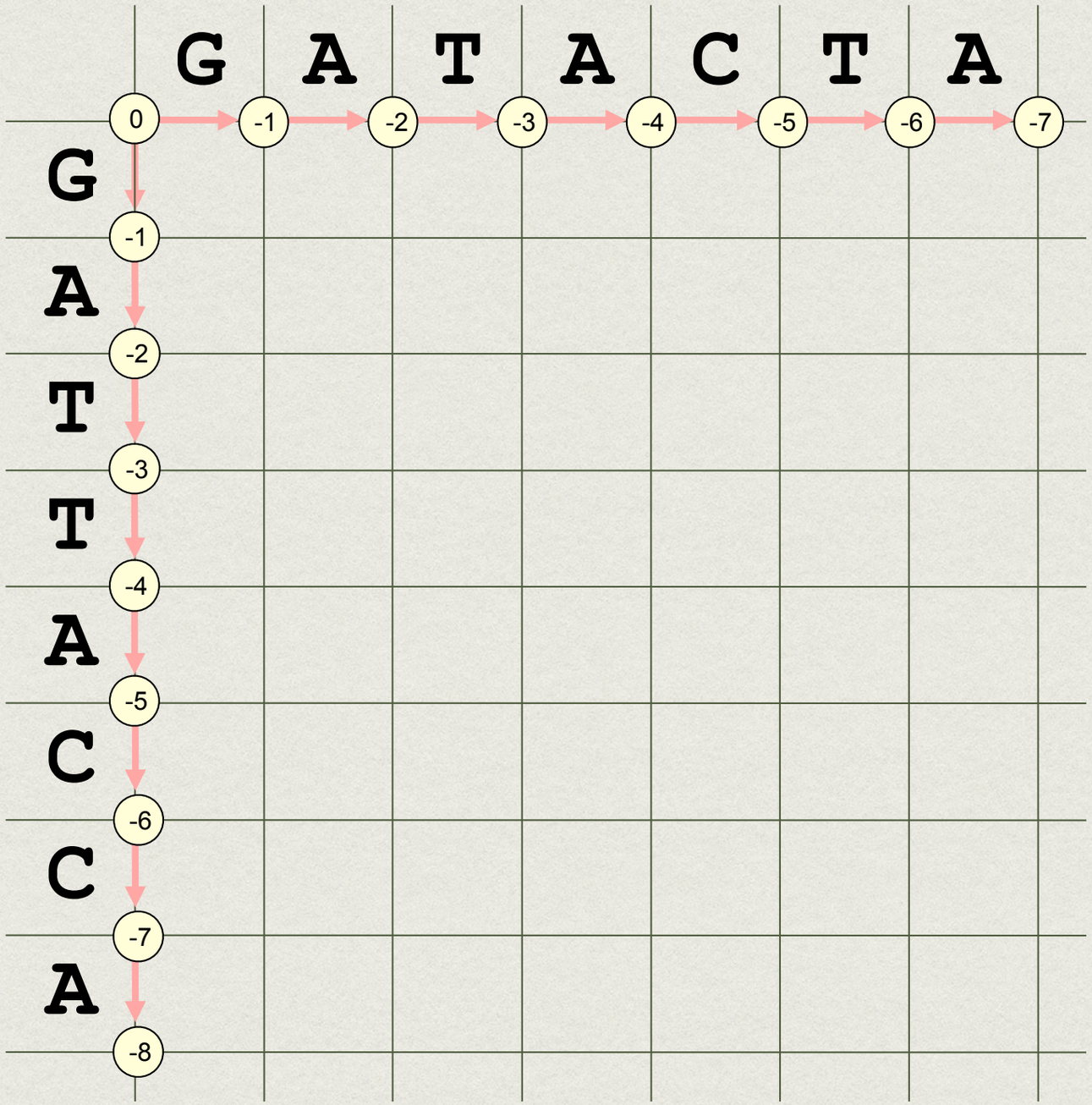
Match: +1

Mismatch: -1

Gap: -1

Dynamic programming

Incrementally extend the path



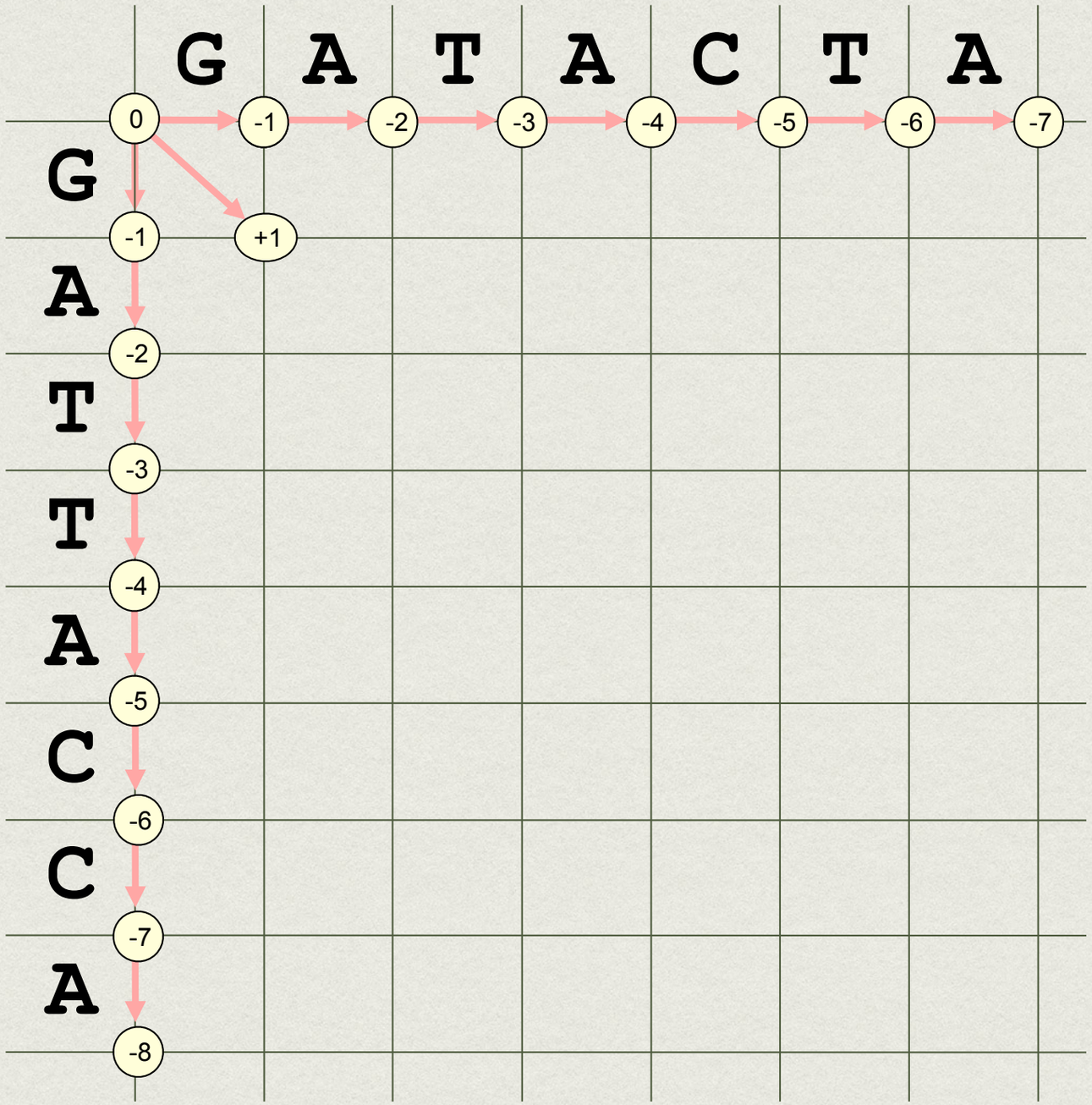
Match: +1

Mismatch: -1

Gap: -1

Dynamic programming

Incrementally extend the path



Match: +1

Mismatch: -1

Gap: -1

Dynamic programming

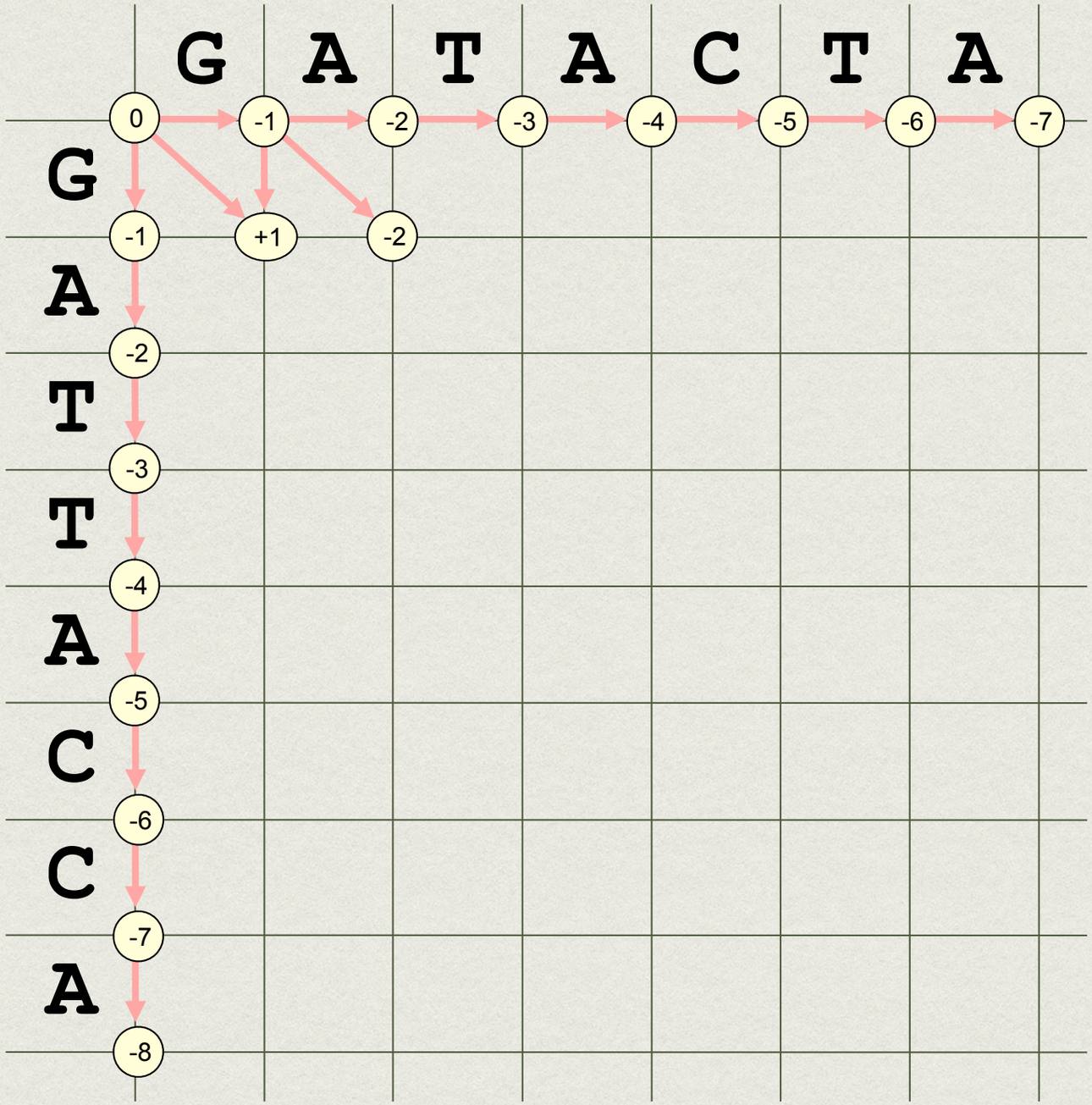
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

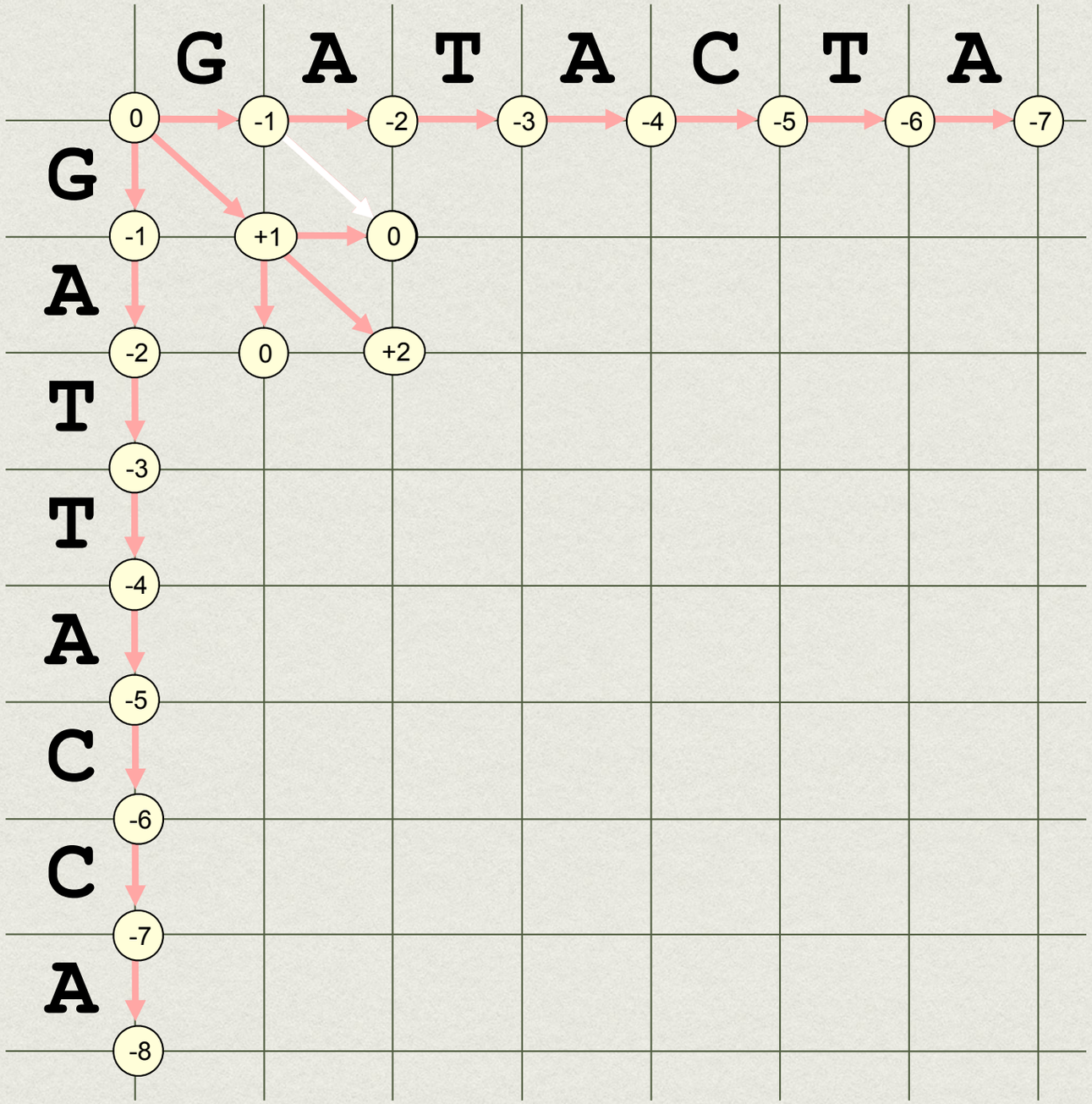
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

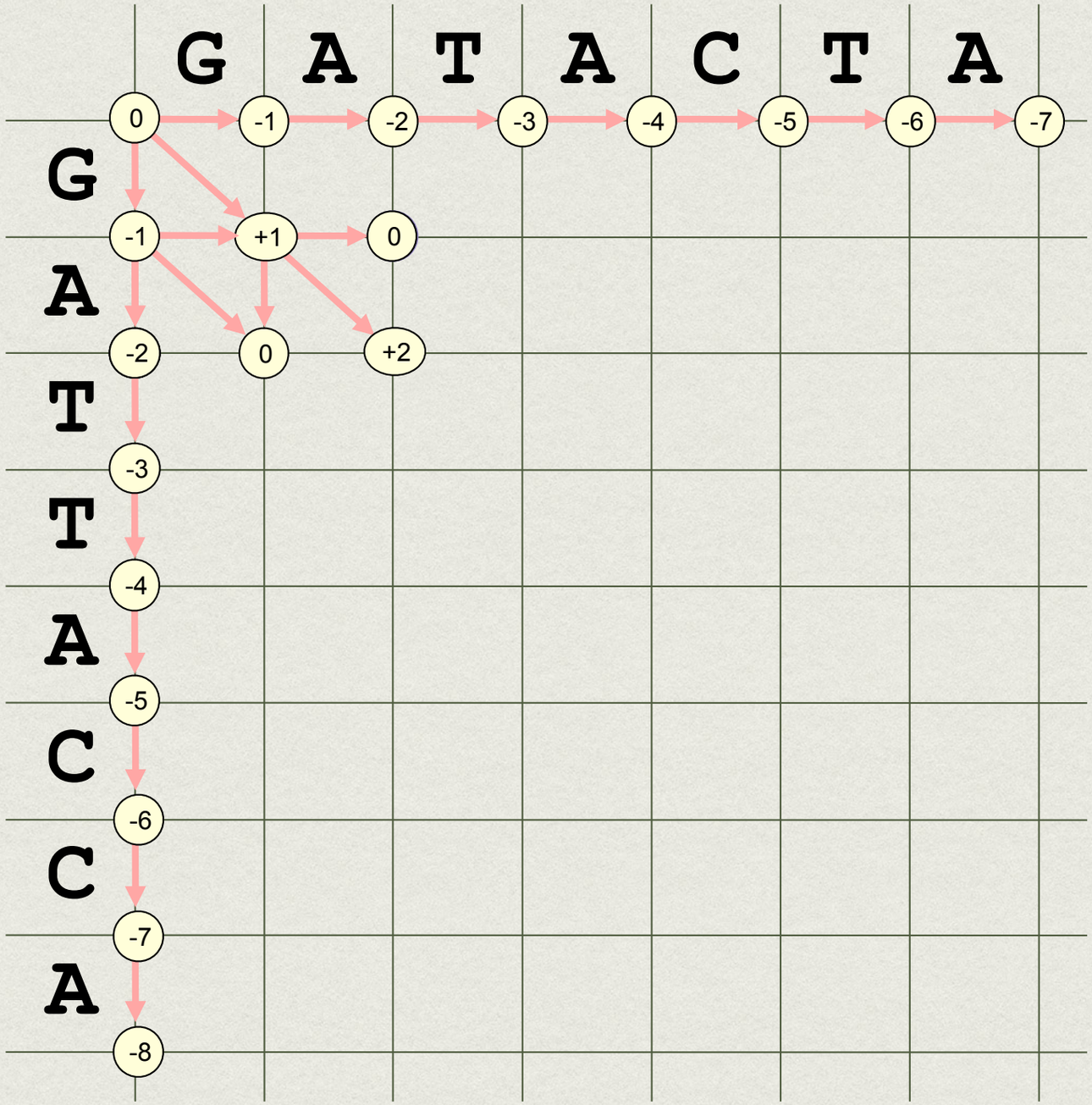
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

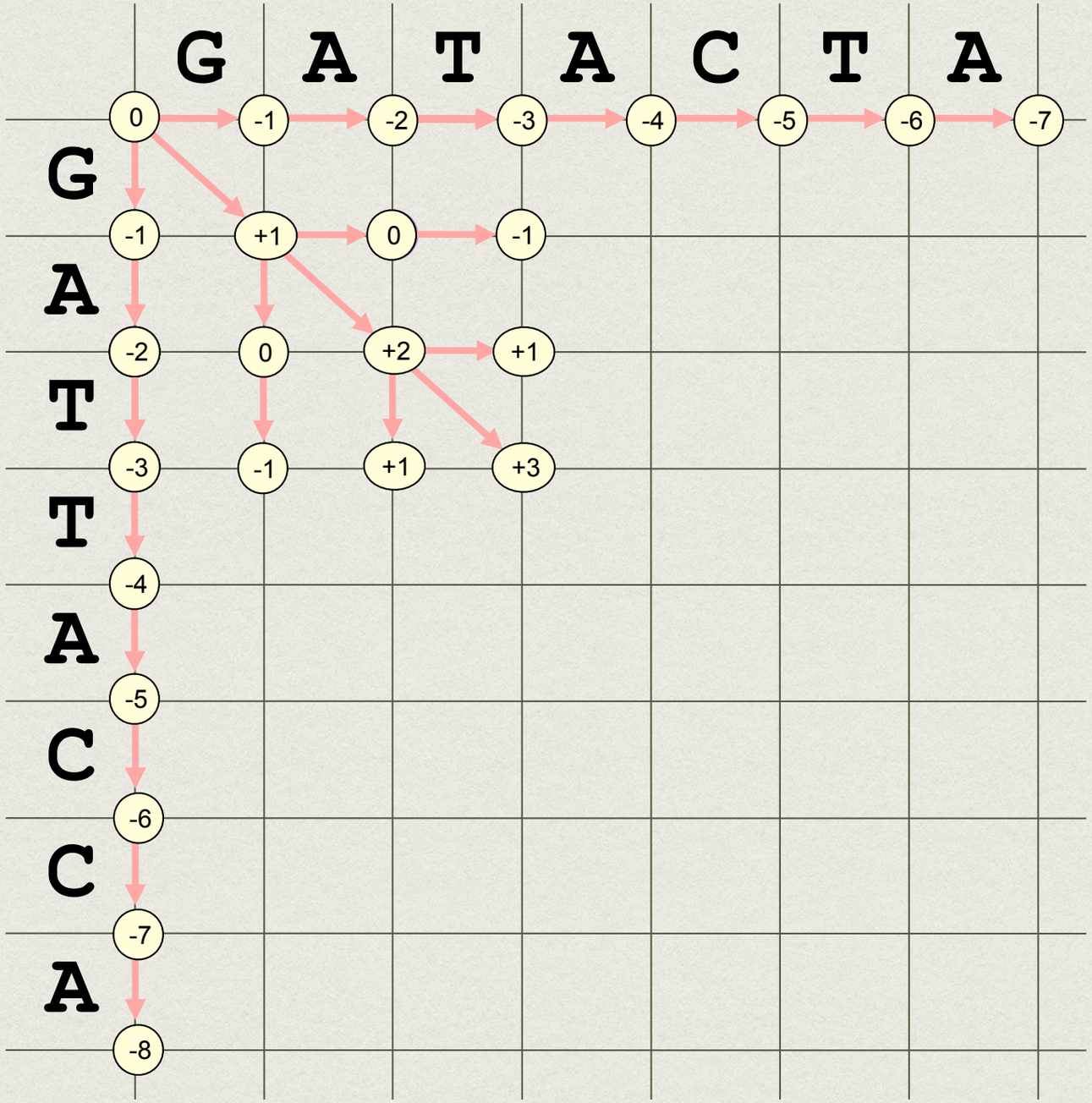
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

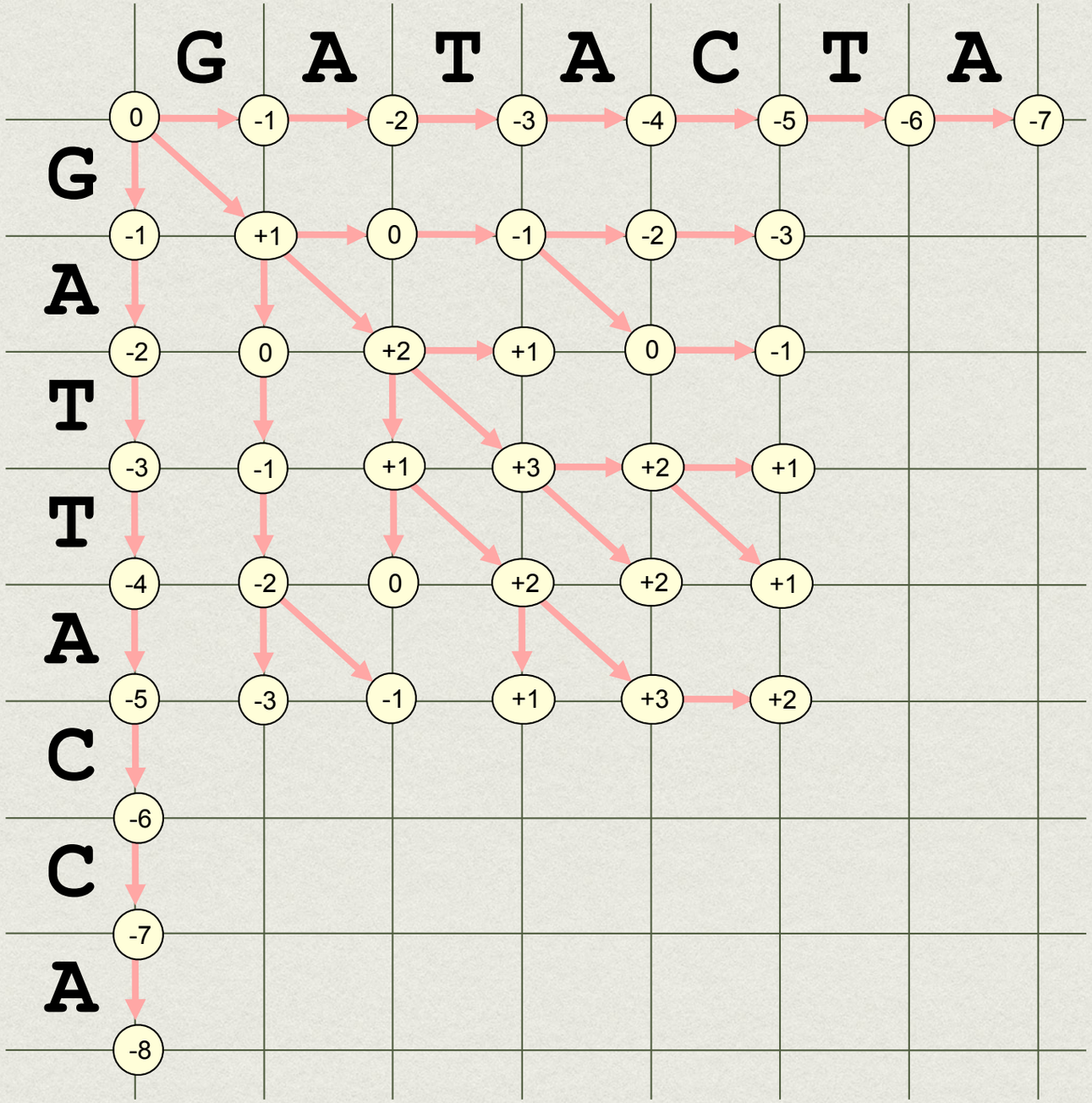
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

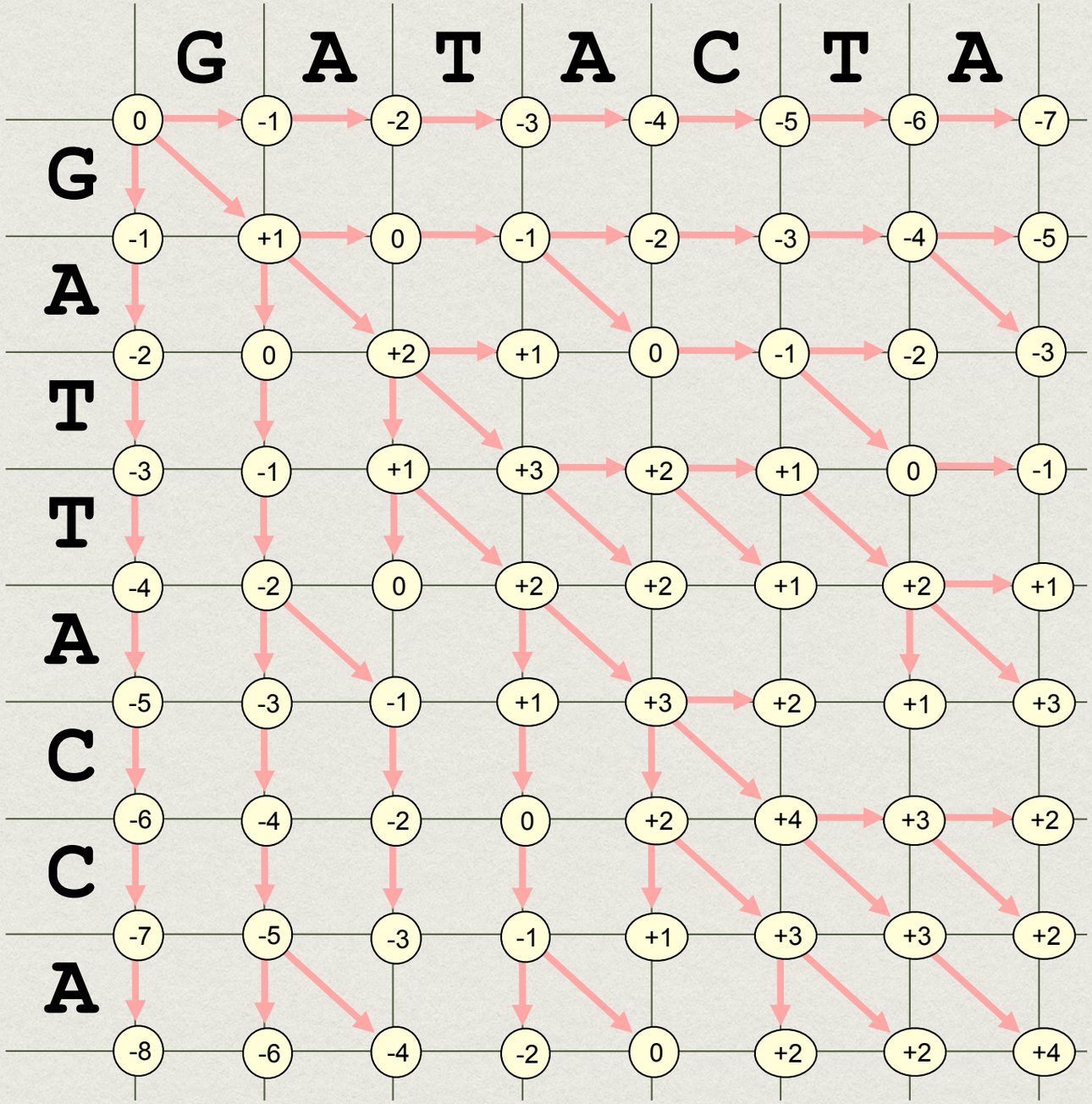
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



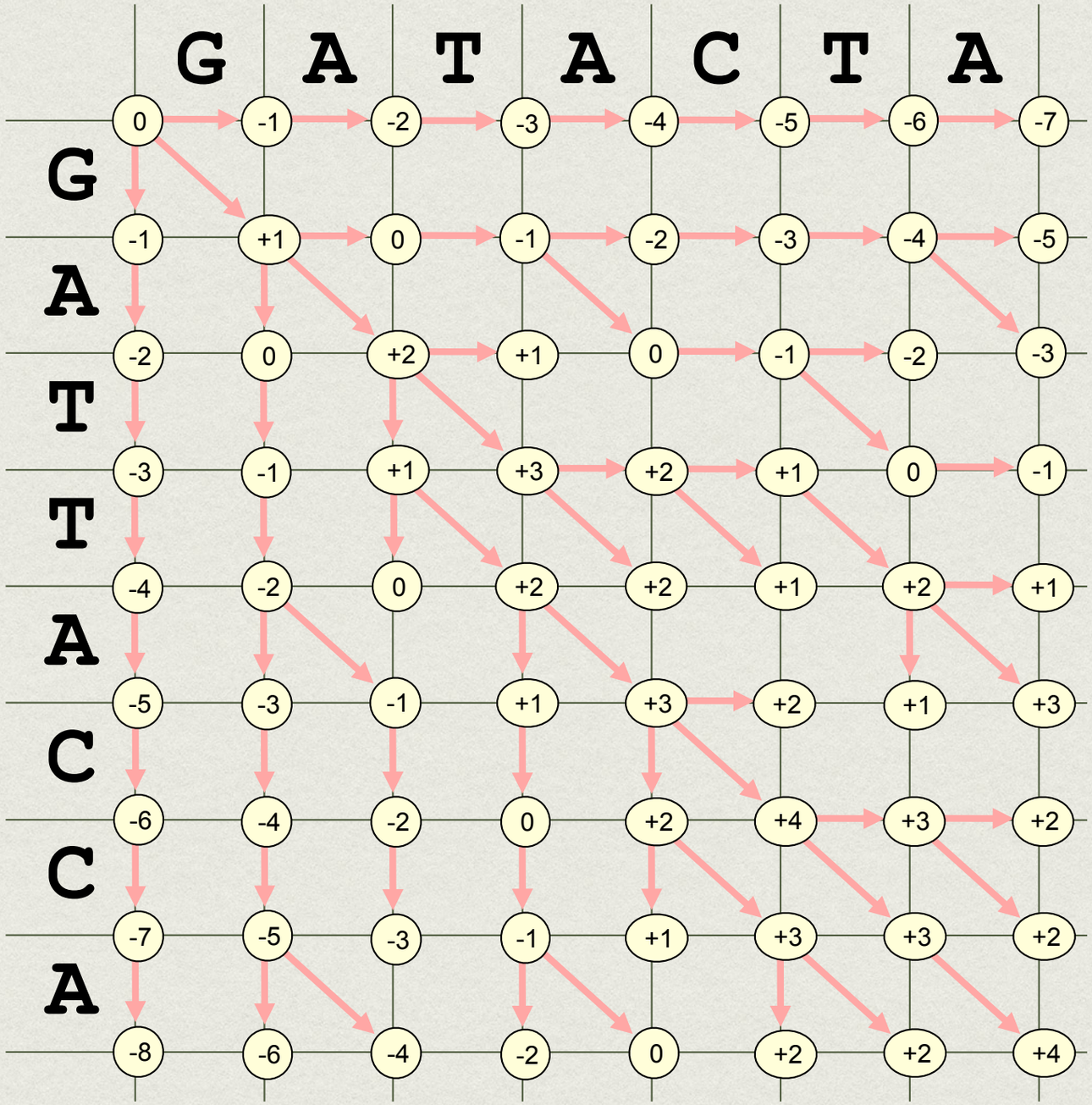
Dynamic programming

Trace back to find optimal alignment

Match: +1

Mismatch: -1

Gap: -1



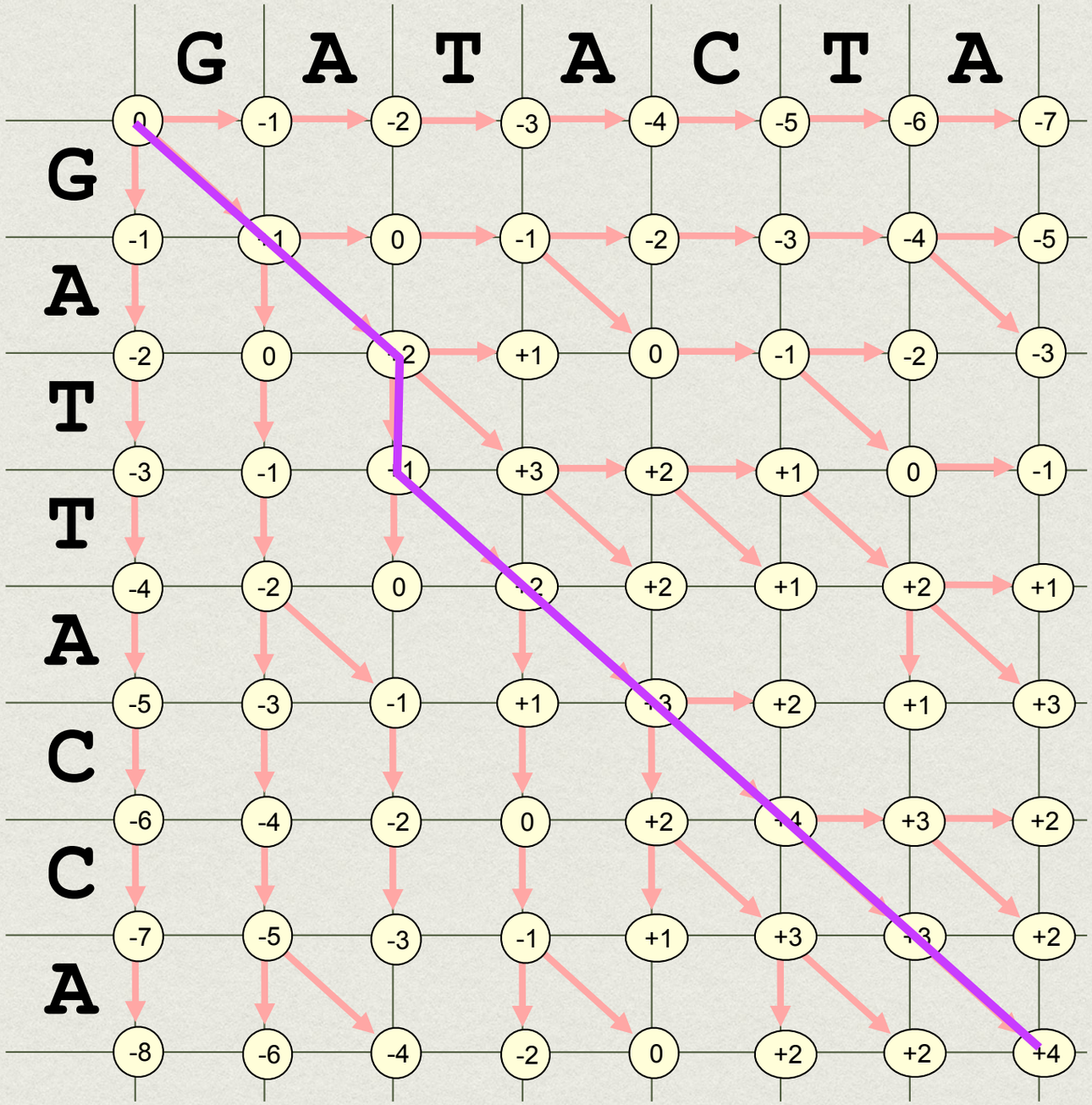
Dynamic programming

Trace back to find optimal alignment

Match: +1

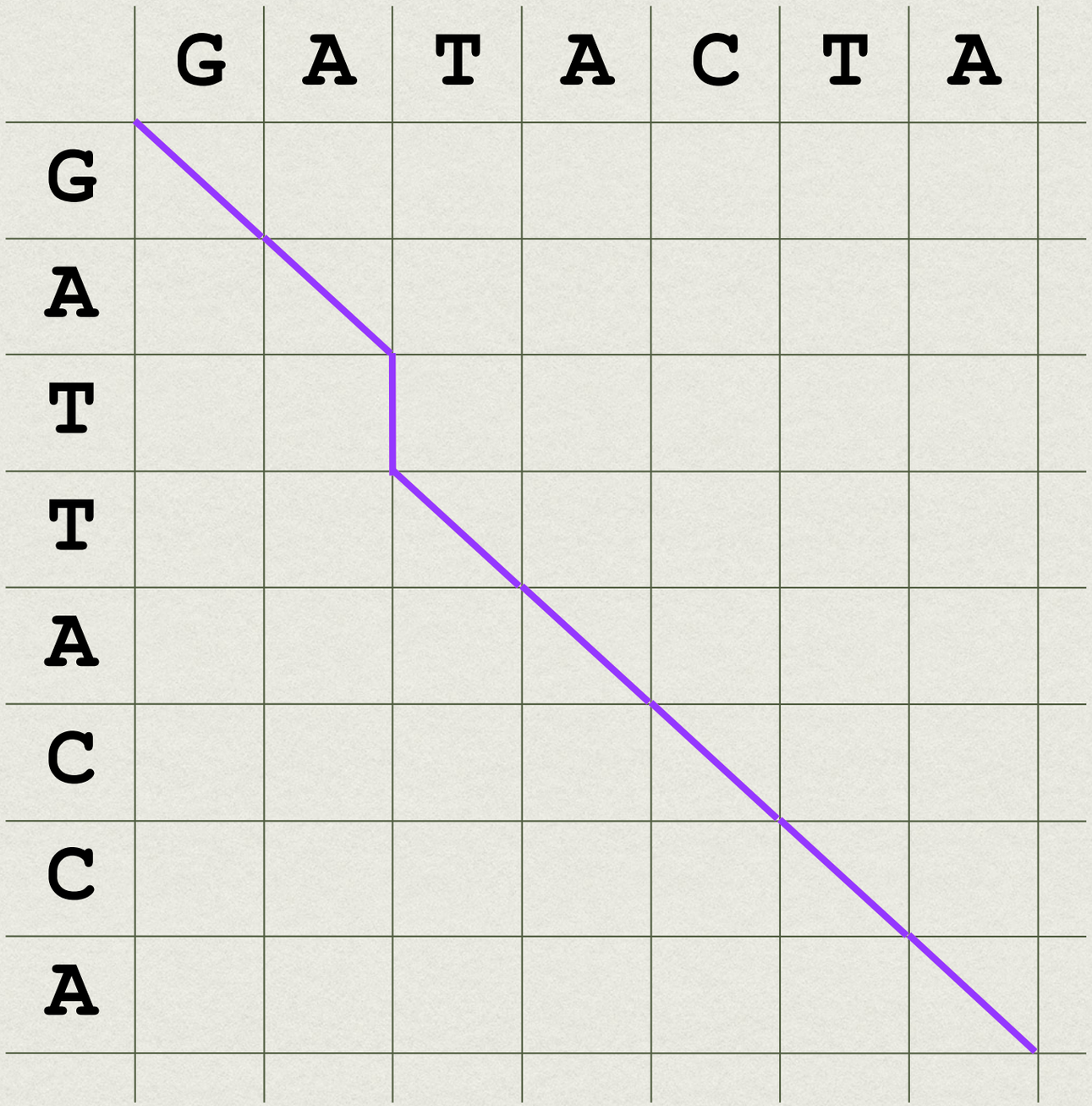
Mismatch: -1

Gap: -1



Dynamic programming

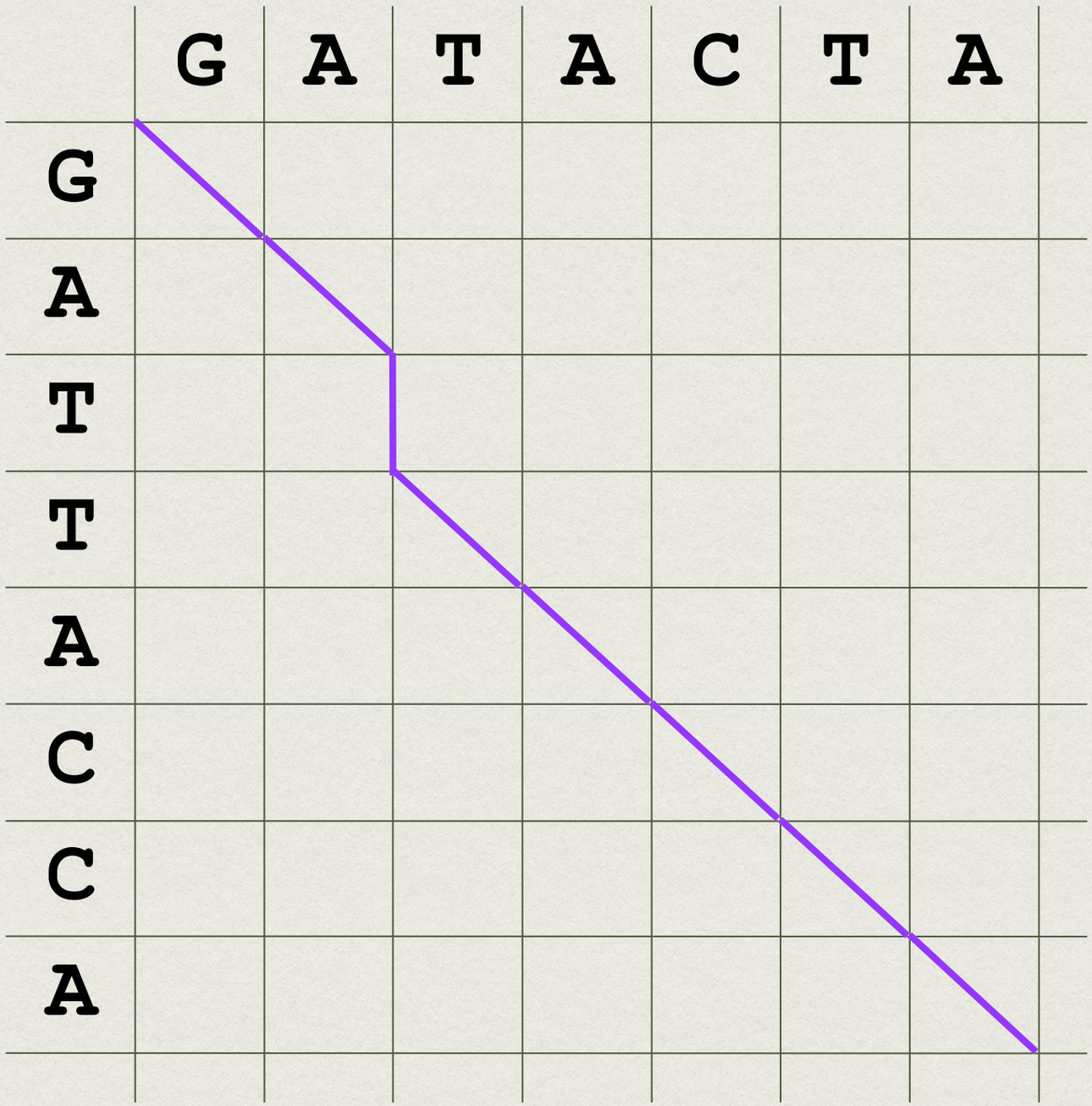
Print out the alignment



Dynamic programming

Print out the alignment

```
GA-TACTA
GATTACCA
```



Dynamic programming

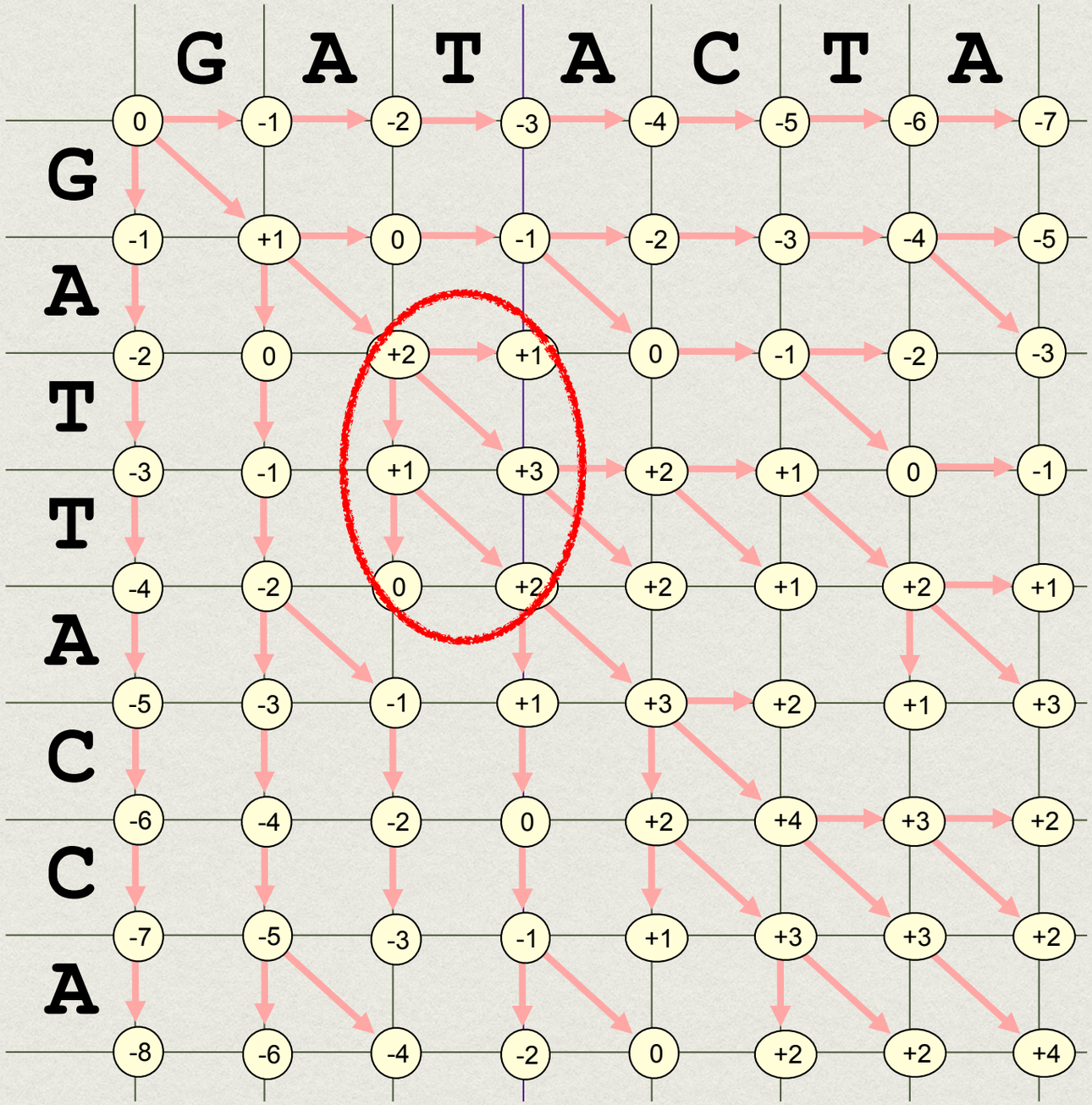
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

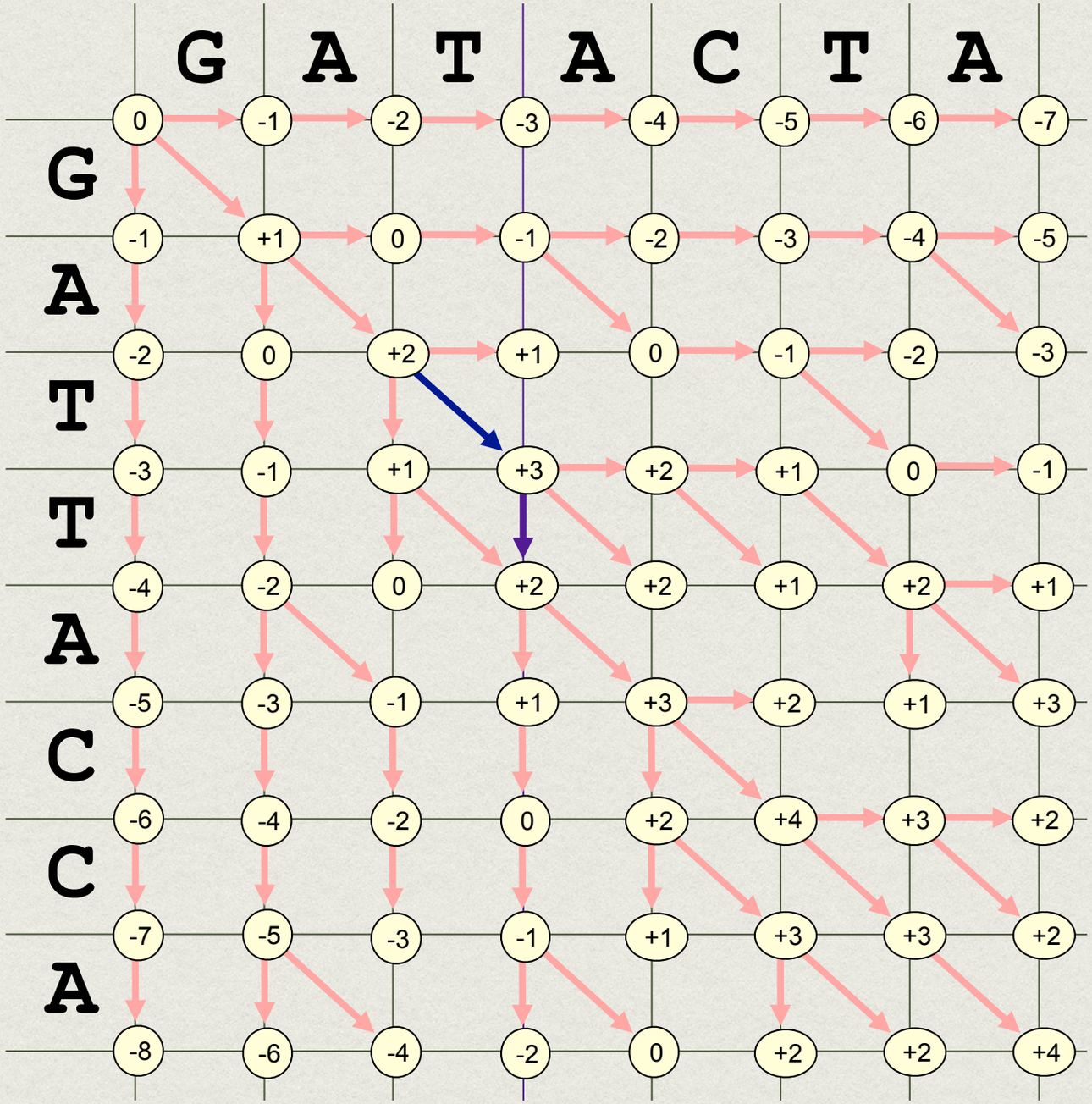
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1



Dynamic programming

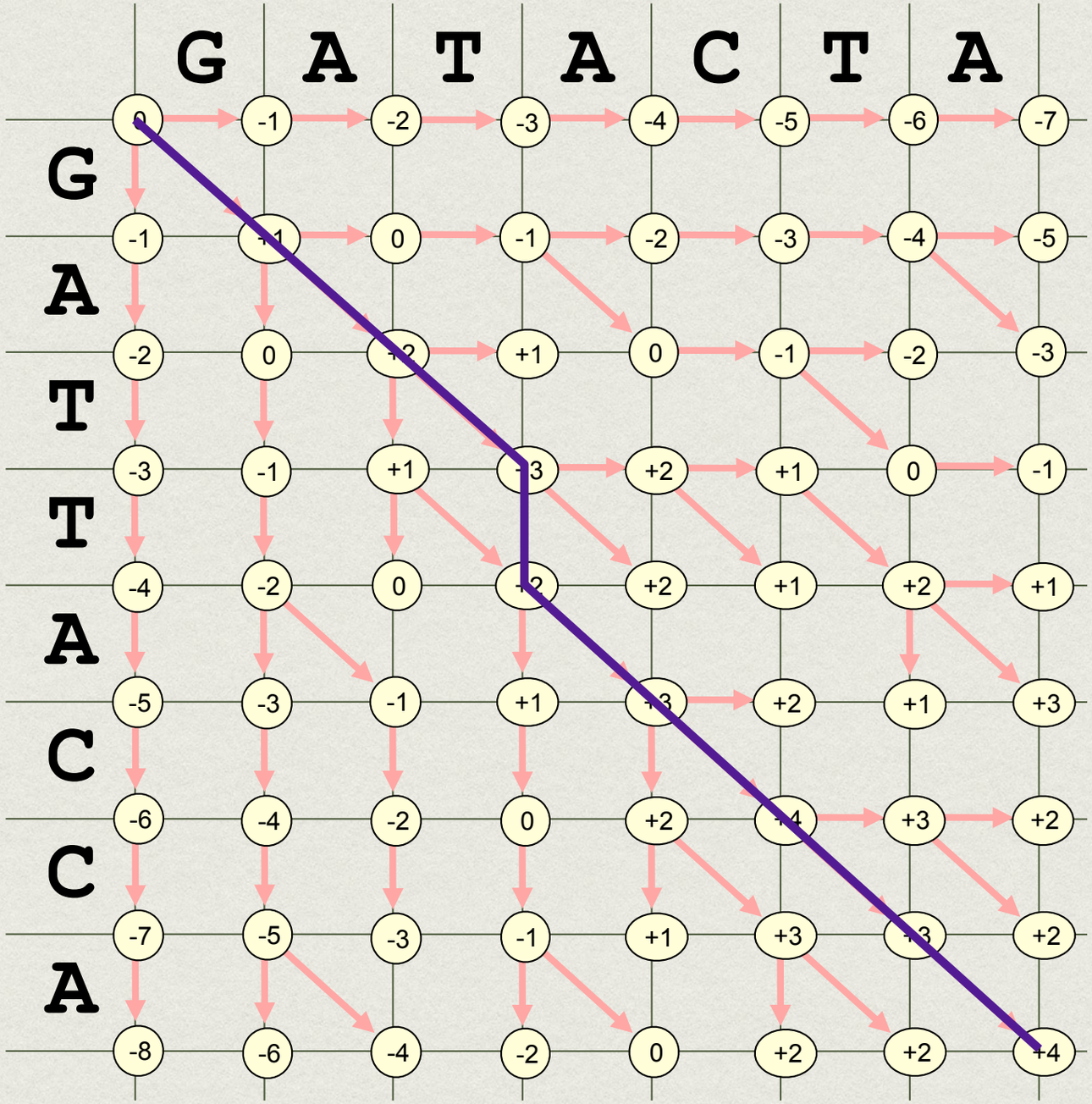
Incrementally extend the path

Remember the best sub-path leading to each point on the lattice

Match: +1

Mismatch: -1

Gap: -1

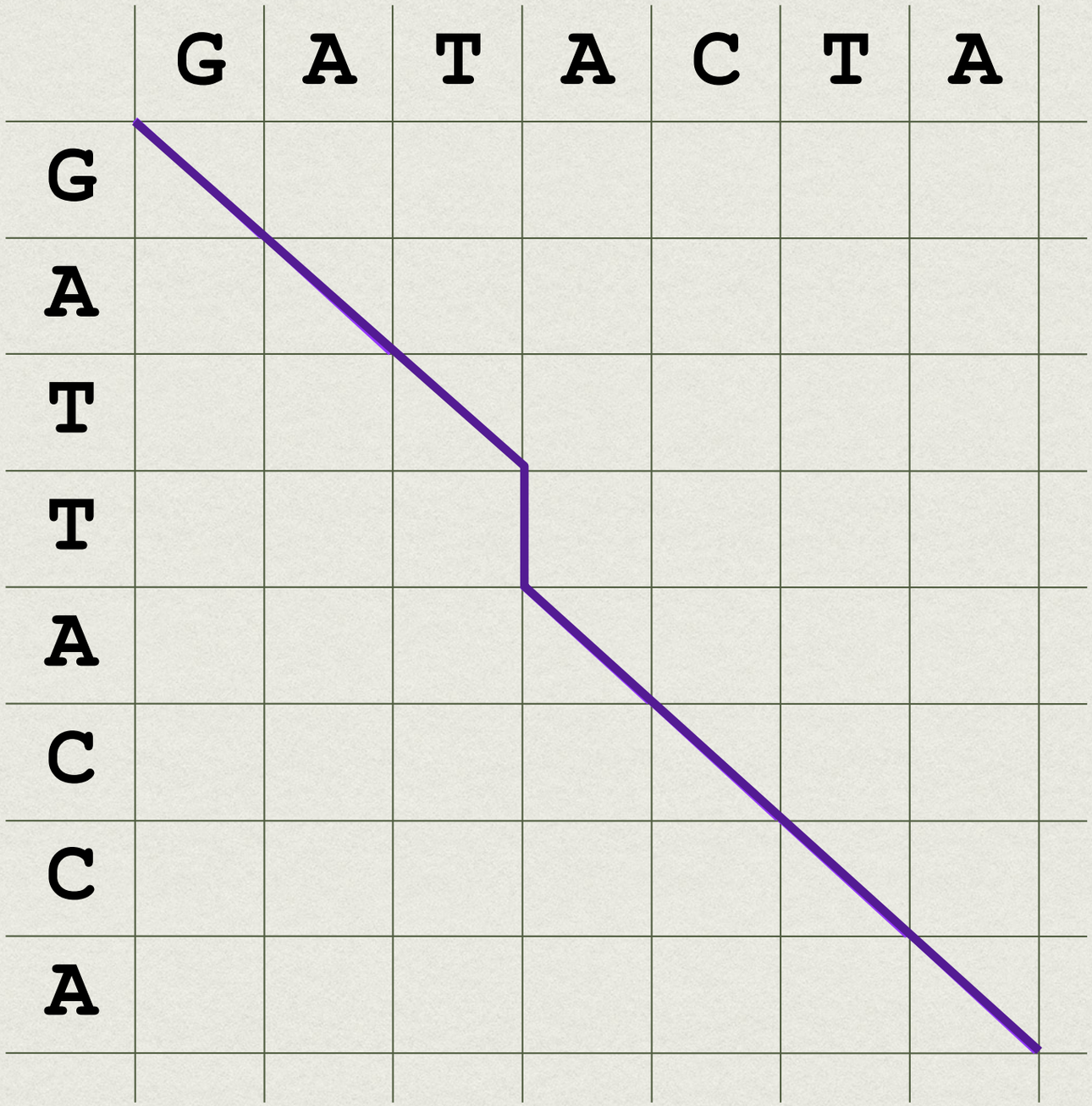


Dynamic programming

Print out the alignment

```
GAT-ACTA
GATTACCA
```

Both alignments are optimal - give the same max. score



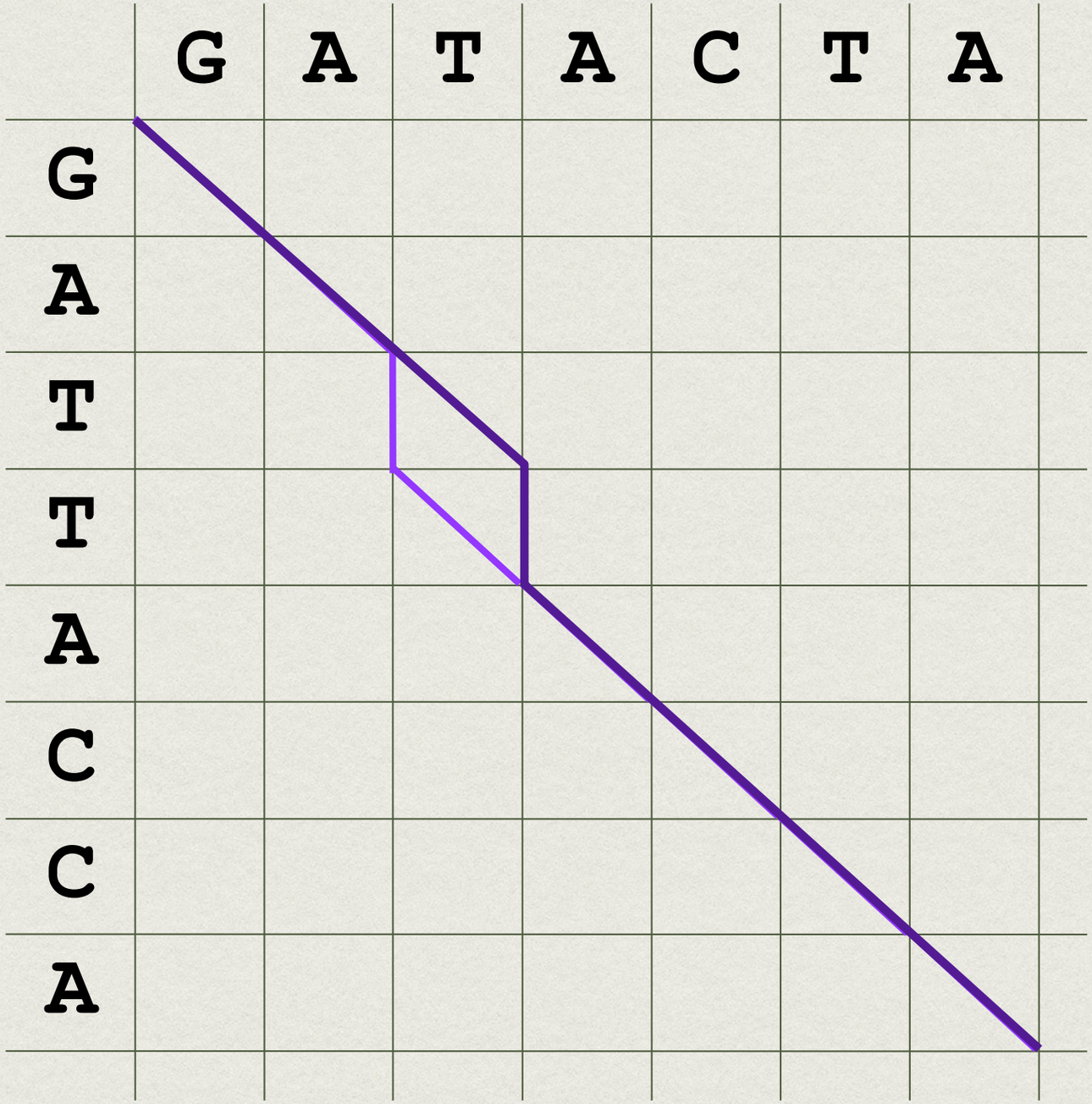
Dynamic programming

Print out the alignment

```
GAT-ACTA
GATTACCA
```

```
GA-TACTA
GATTACCA
```

Both alignments are optimal - give the same max. score



AMINO ACID SCORING SYSTEMS

- more complicated than nucleotide matrices
- first, we can align two homologous protein sequences and count the number of any particular substitution, for instance Serine to Threonine
- a likely change should score higher than a rare one
- we have to take into account that several the same position mutated several times after sequence divergence - this could bias statistics

AMINO ACID SCORING SYSTEMS

- to avoid this problem one can compare very similar sequences so one can assume that no position has changed more than once

- Margret Dayhoff introduced the PAM system (Percent of Accepted Mutations)

  - 1 PAM - two sequence have 99% identical residues
  - 10 PAM - two sequence have 90% identical residues

APPROXIMATE RELATION BETWEEN PAM AND SEQUENCE IDENTITY

PAM	0	30	80	110	200	250
AA sequence identity (%)	100	75	50	40	25	20

PAM matrix is expressed as log-odds values multiplied by 10 simply to avoid decimal points

PAM MATRIX CALCULATION

$$\text{score of substitution } i <-> j = \log \frac{\text{observed } i <-> j \text{ mutation rate}}{\text{mutation rate expected from amino acids frequencies}}$$

For instance, a value 2 implies that in related sequences the mutation would be expected to occur 1.6 times more frequently than random.

The calculation: The matrix entry 2 corresponds to the actual value 0.2 because of the scaling. The value 0.2 is $\log_{10}$ of the relative expectation value of the mutation. Therefore, the expectation value is $10^{0.2} = 1.6$

AMINO ACID MATRICES

- Problem with PAM schema lies in that the high number matrices are extrapolated from closely related sequences
- Henikoffs developed the family of BLOSUM matrices based on the BLOCKS database of aligned protein sequences, hence the name BLOcks SUbstitution Matrix
- observed substitution frequencies taken from conserved regions of proteins (blocks), not the whole proteins as in case of Dayhoff's work
- to avoid overweighting closely related sequences, the Hennikoffs replaced groups of proteins that have sequence identities higher than a threshold by either a single representative or a weighted average, e.g. for the commonly used BLOSUM62 matrix the threshold is 62%
- NOTE reversed numbering of PAM and BLOSUM matrices

BLOSUM 62 SCORING MATRIX

```
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
```

some replacement are more frequent than others

score system based on comparison of homologous domains

BLOSUM 62 SCORING MATRIX

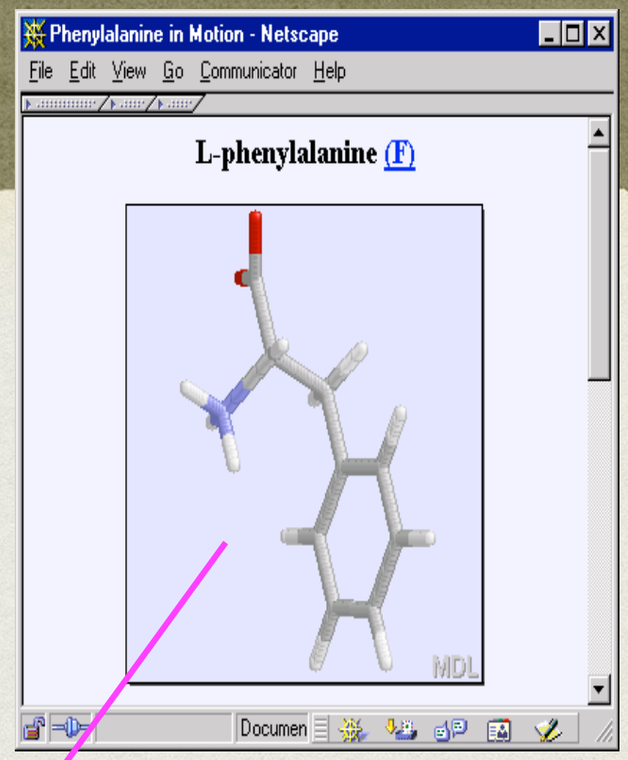
```
A  4
R -1  5
N -2  0  6
D -2 -2  1  6
C  0 -3 -3 -3  9
Q -1  1  0  0 -3  5
E -1  0  0  2 -4  2  5
G  0 -2  0 -1 -3 -2 -2  6
H -2  0  1 -1 -3  0  0 -2  8
I -1 -3 -3 -3 -1 -3 -3 -4 -3  4
L -1 -2 -3 -4 -1 -2 -3 -4 -3  2  4
K -1  2  0 -1 -3  1  1 -2 -1 -3 -2  5
M -1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  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
```

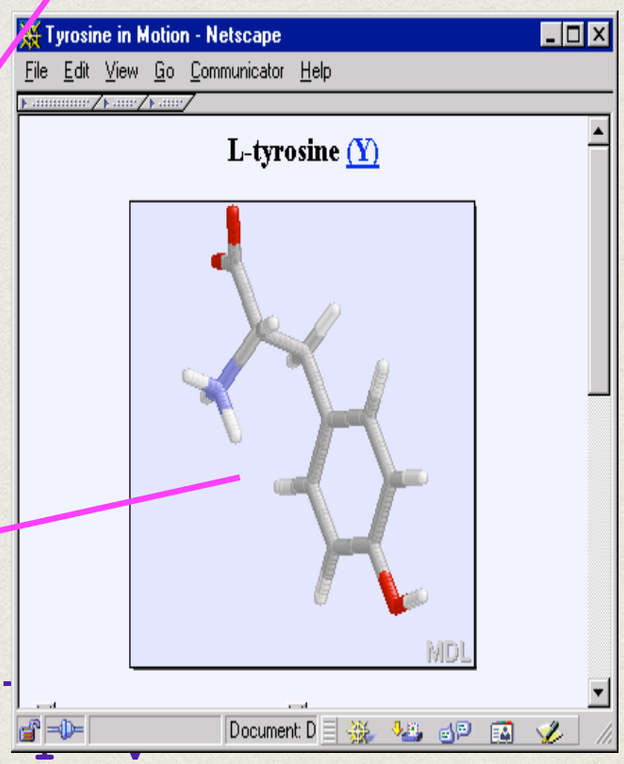
identical amino acids get positive score but not the same - frequent amino acids, e.g. alanine, get lower score than rare amino acids, e.g. tryptophan

BLOSUM 62 SCORING MATRIX

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

substitutions to amino acids of similar properties give a positive score

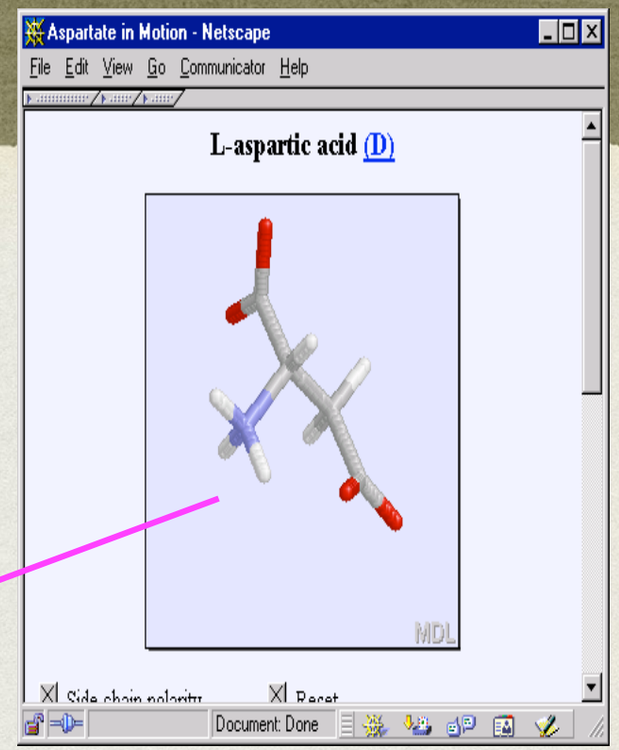


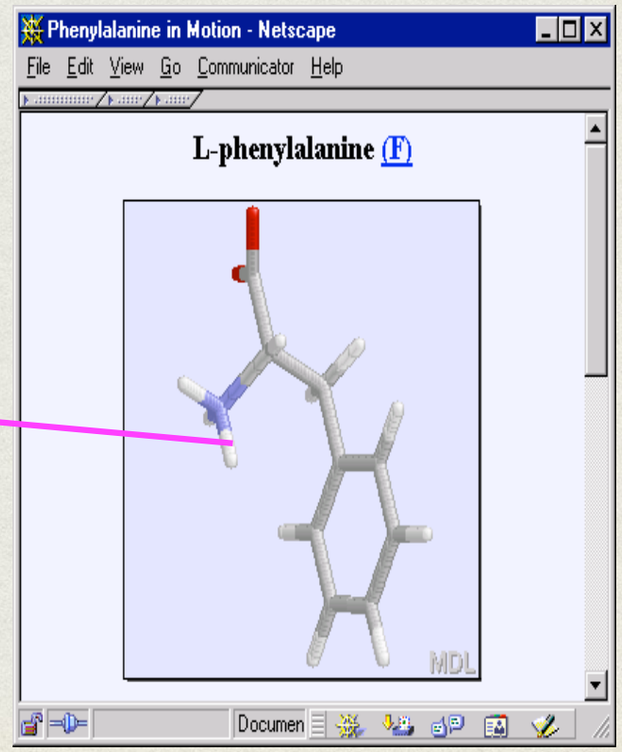


BLOSUM 62 SCORING MATRIX

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

substitutions to amino acids of different properties give a negative score





SCORING RECOMMENDATIONS

- nucleotide sequence comparison
  - match +10, mismatch -3, gap opening -50, gap extension -5
- amino acid sequence comparison
  - for general use (e.g. unknown sequence similarity) - BLOSUM62
  - for diverged proteins - PAM250 or BLOSUM30
  - for similar sequences - PAM15 or BLOSUM80

SEQUENCE SIMILARITY SEARCH

BASICS OF DATABASE SEARCH

- Database searching is fundamentally different from alignment
- The goal is to find homologous sequences (often more than one), not to establish the correct one-to-one mapping of particular residues
- Usually, this is a necessary first step to making an information map between two sequences
- Database searching programs were originally thought of as approximations to dynamic programming alignments
- Assumption: the best database search conditions are those that would produce the “correct” alignment
- Key idea - most sequences don’t match. If one can find a fast way to eliminate sequences that don't match, the search will go much faster

BASICS OF DATABASE SEARCH

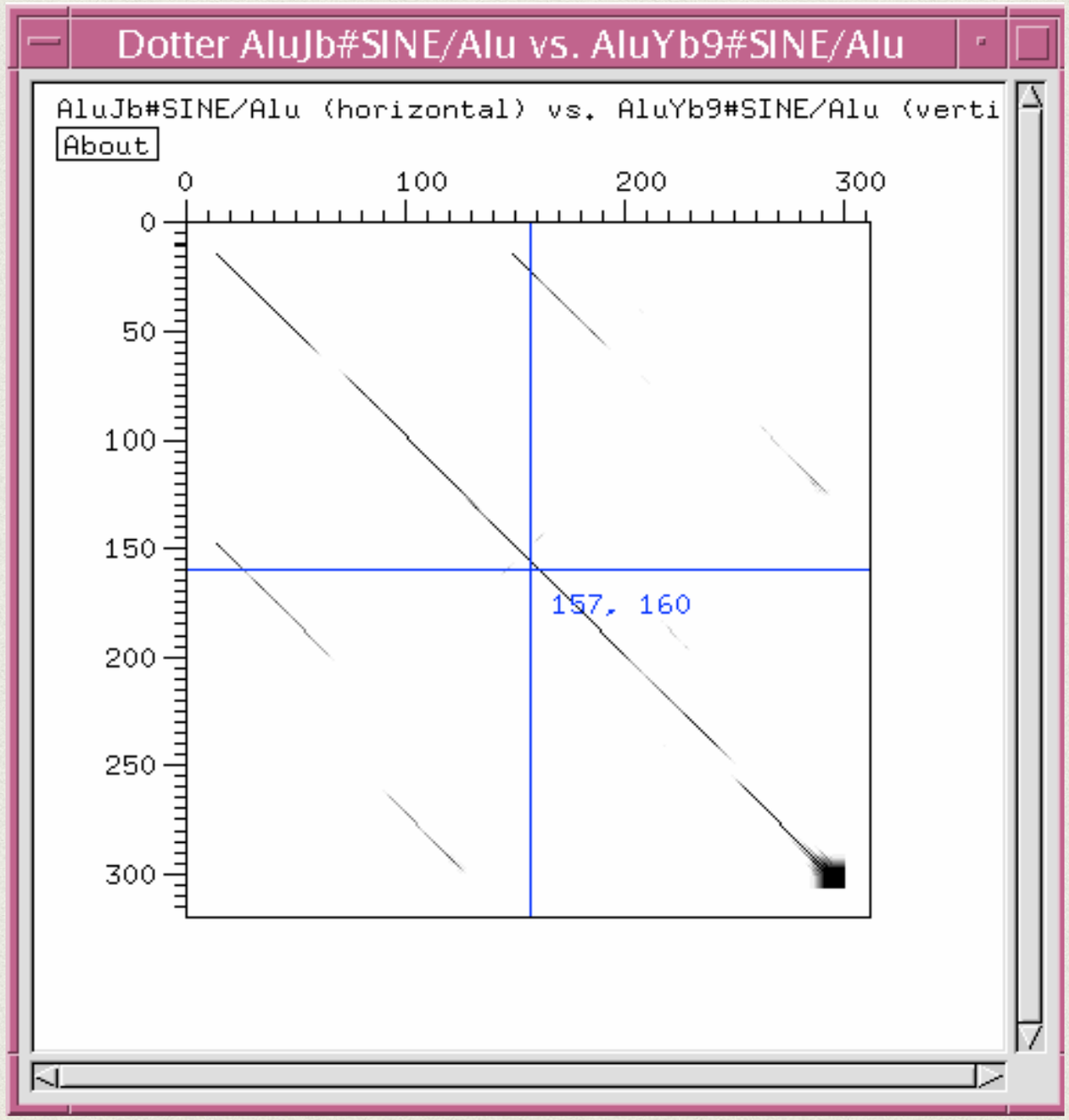
basic terminology:

query - sequence to be used for the database search

subject - sequence found in the database that meets some similarity criteria

hit - local alignment between query and subject

Related sequences have "diagonals" with high similarity

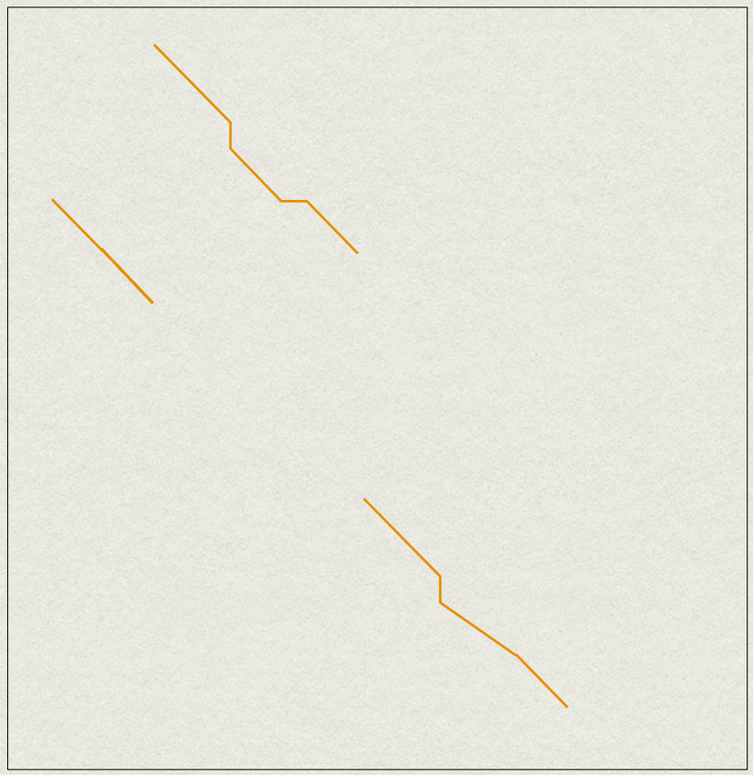


GLOBAL VERSUS LOCAL ALIGNMENT

Optimal global alignment

Sequences align essentially from end to end.

Optimal local alignment



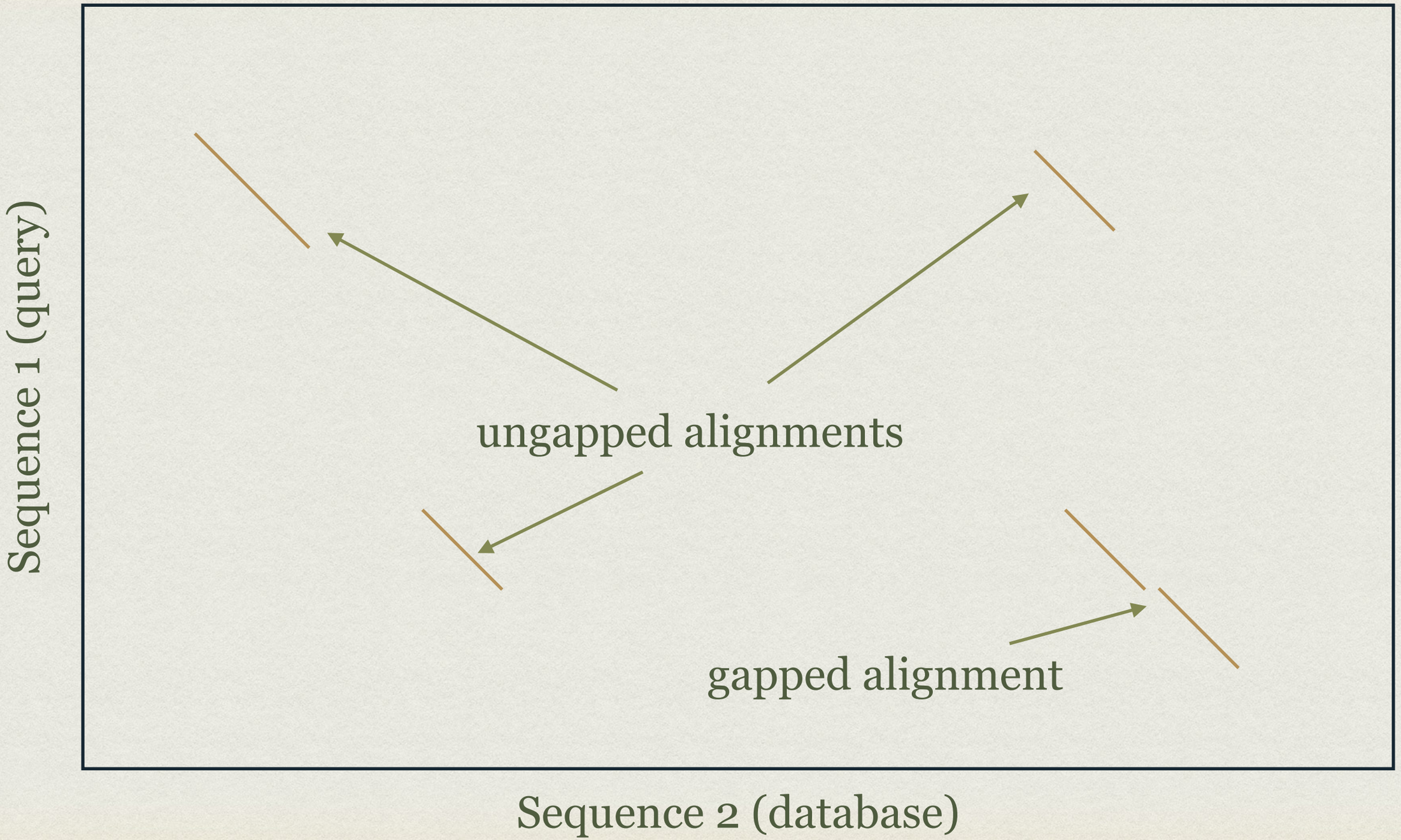
Sequences align only in small, isolated regions.

BLAST

Basic Local Alignment Search Tool

References:
Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410.
Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. (1997) "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic Acids Res. 25:3389-3402

BLAST ALGORITM - SEARCH SPACE



NUCLEOTIDE BLAST ALGORITHM

1. Break down query sequence into overlapping words (*k*-mers).

```
TAATTGCGCTAGGATTCGCTAAT
TAA     GCT     TTC
 AAT     CTA     TCG
  ATT     TAG     CGC
   TTG     AGG     GCT
    TGC     GGA     CTA
     GCG     GAT     TAA
      CGC     ATT     AAT
```

NUCLEOTIDE BLAST ALGORITHM

1. Break down query sequence into overlapping words.

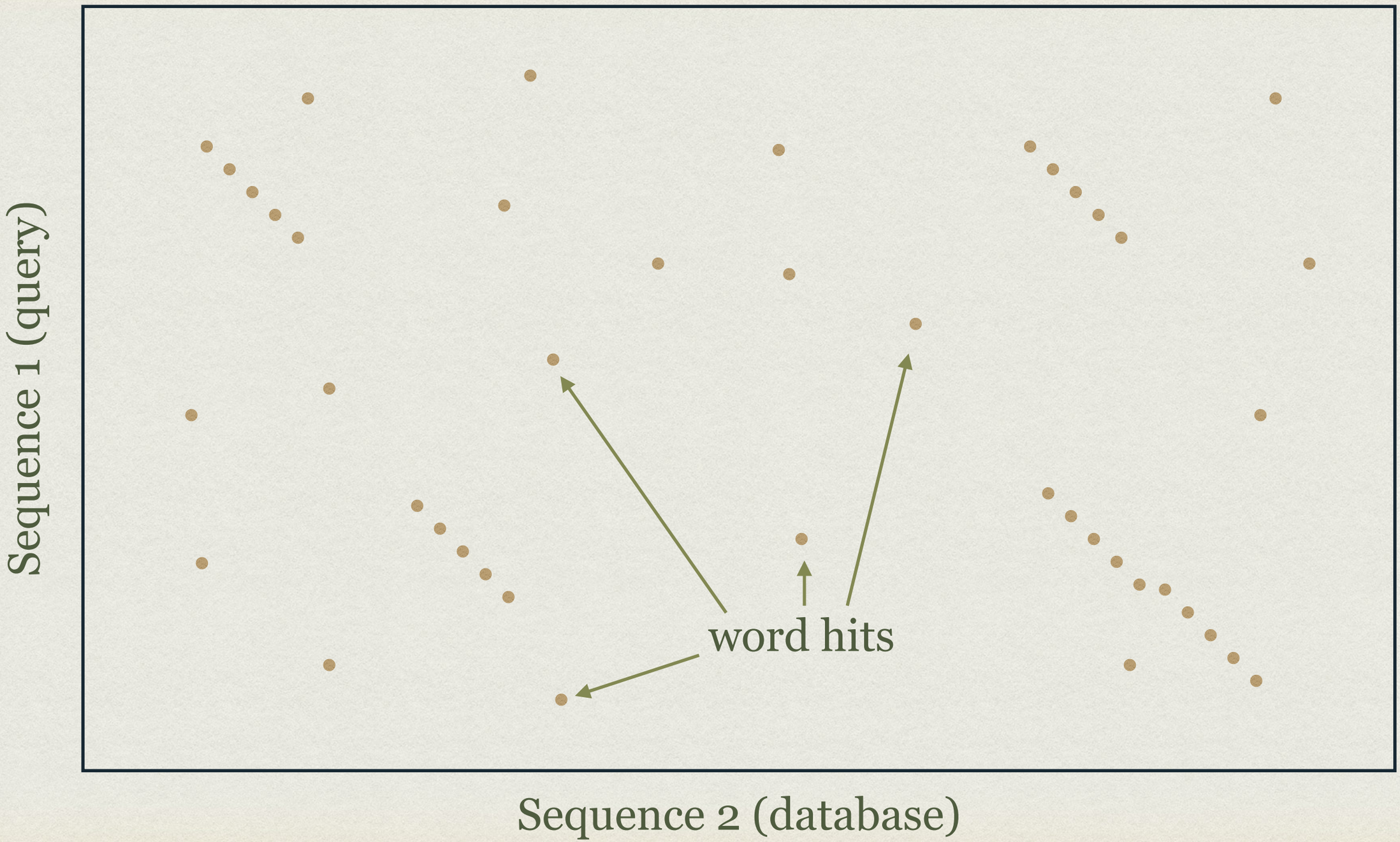
```
TAATTGCGCTAGGATTCGCTAAT

TAA     GCT     TTC
 AAT     CTA     TCG
  ATT     TAG     CGC
   TTG     AGG     GCT
    TGC     GGA     CTA
     GCG     GAT     TAA
      CGC     ATT     AAT
```

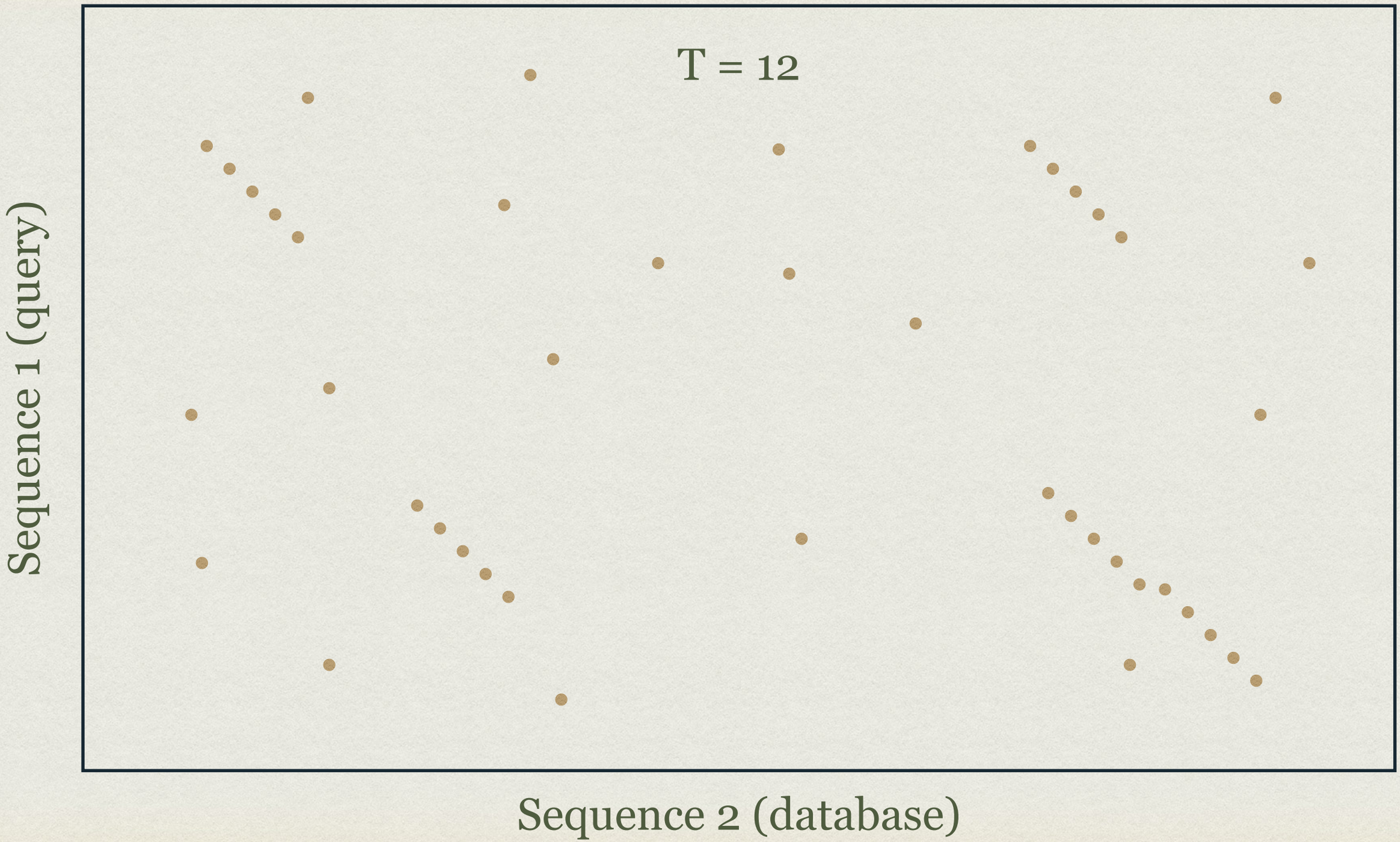
NUCLEOTIDE BLAST ALGORITHM

1. Break down query sequence into overlapping words.
2. Scan databases for exact matches of size W (BLASTn) or 110110 pattern (MegaBlast).

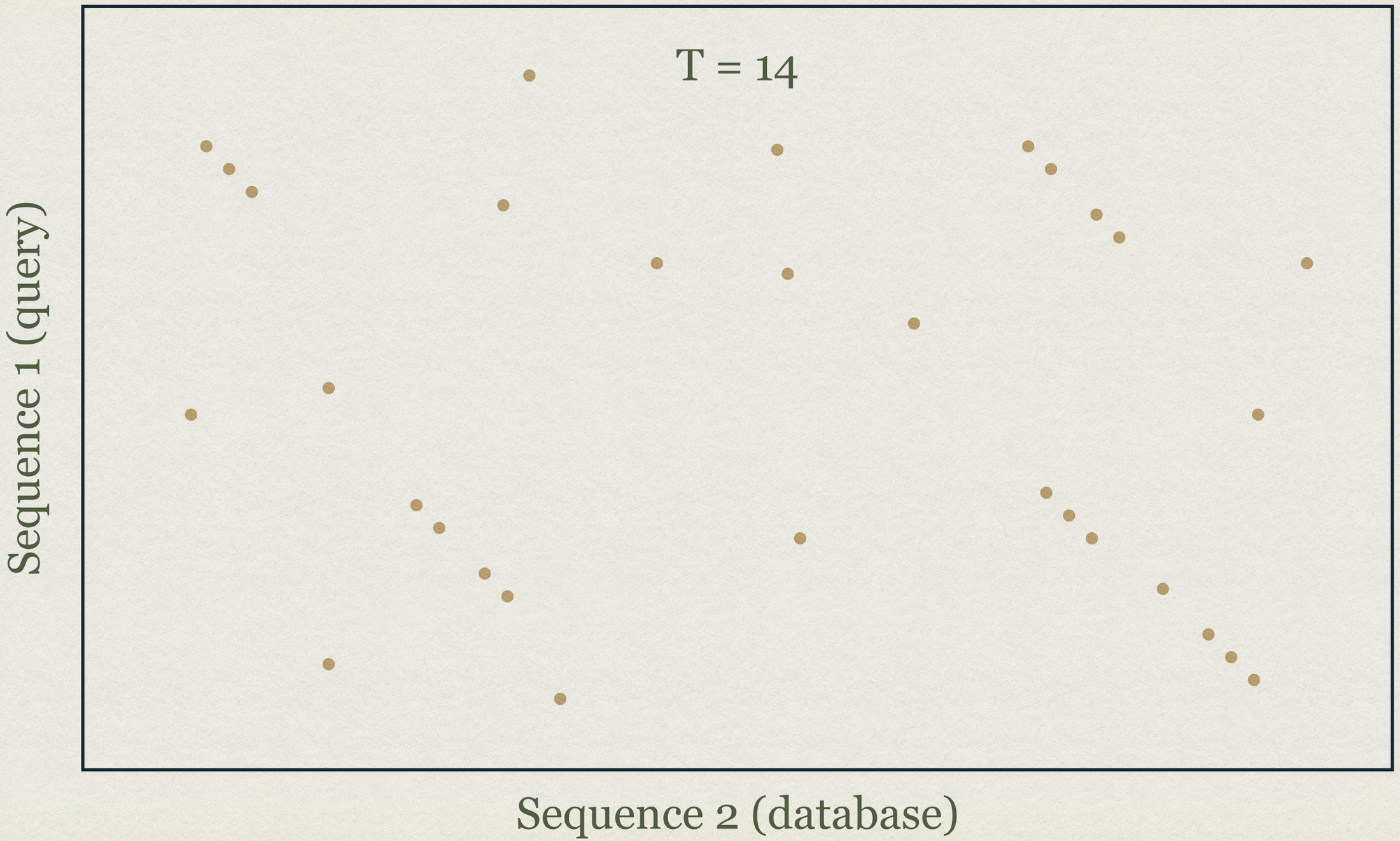
BLAST ALGORITM - SEARCH SPACE



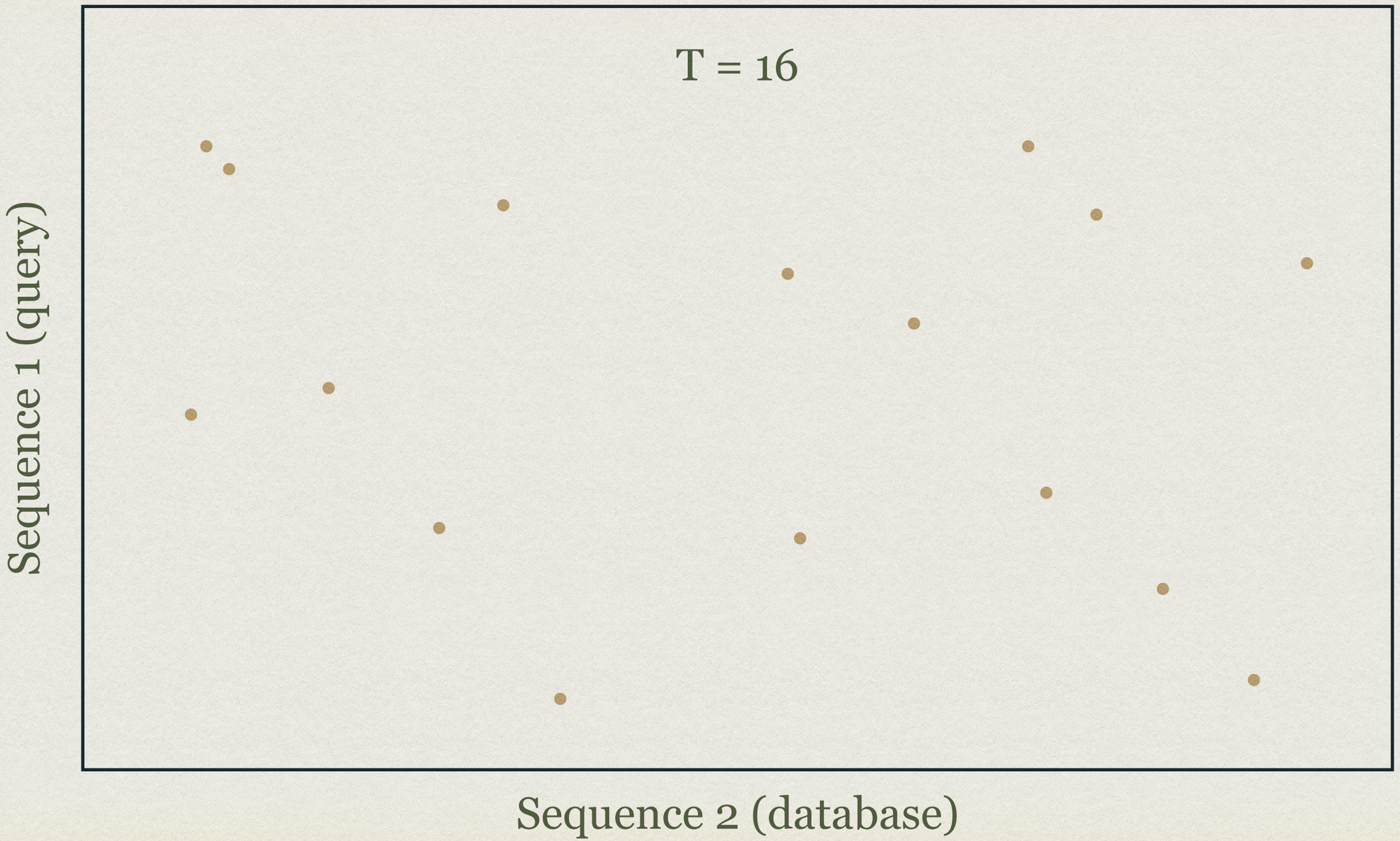
BLAST ALGORITM - SEARCH SPACE



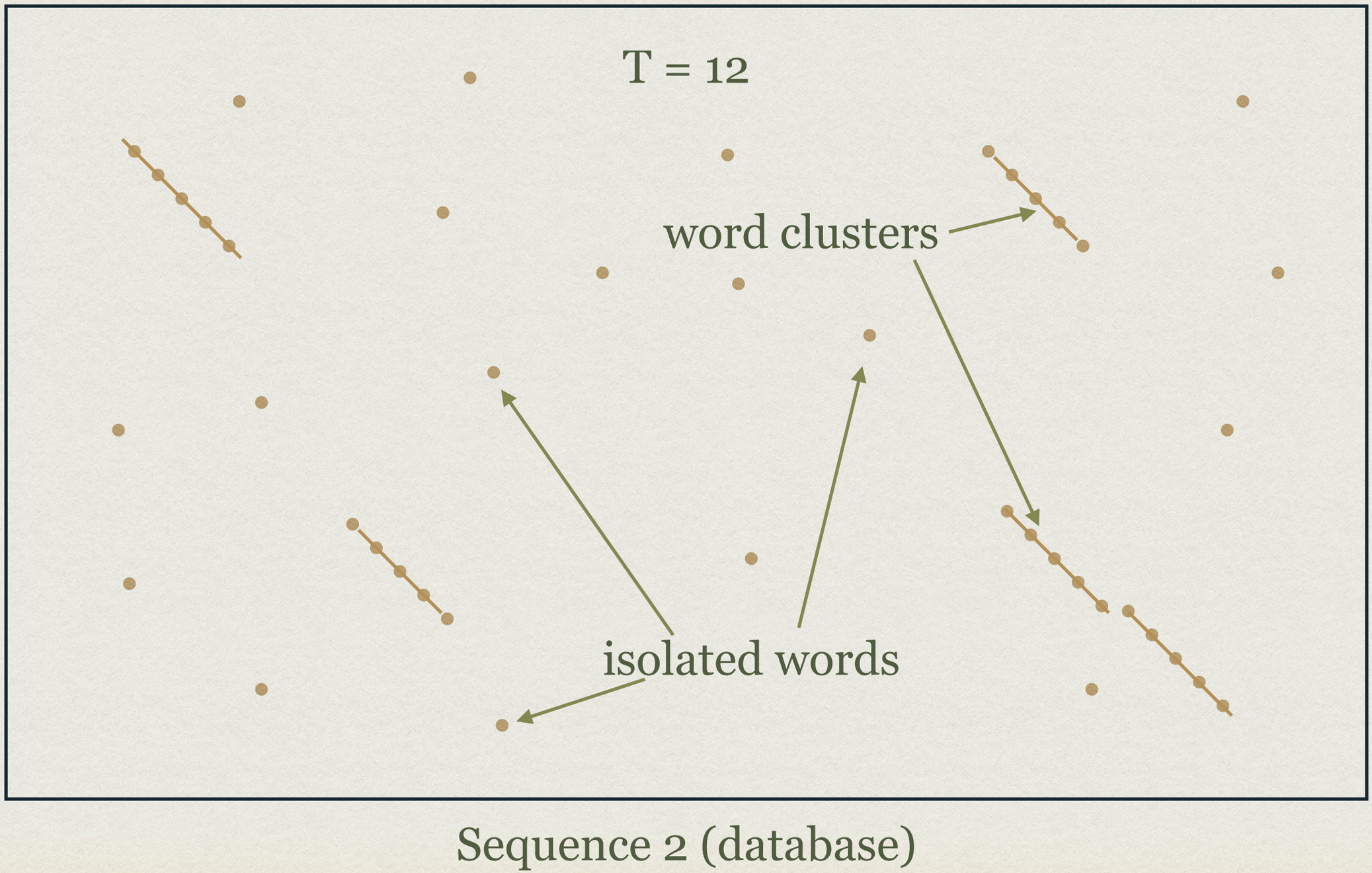
BLAST ALGORITM - SEARCH SPACE



BLAST ALGORITM - SEARCH SPACE



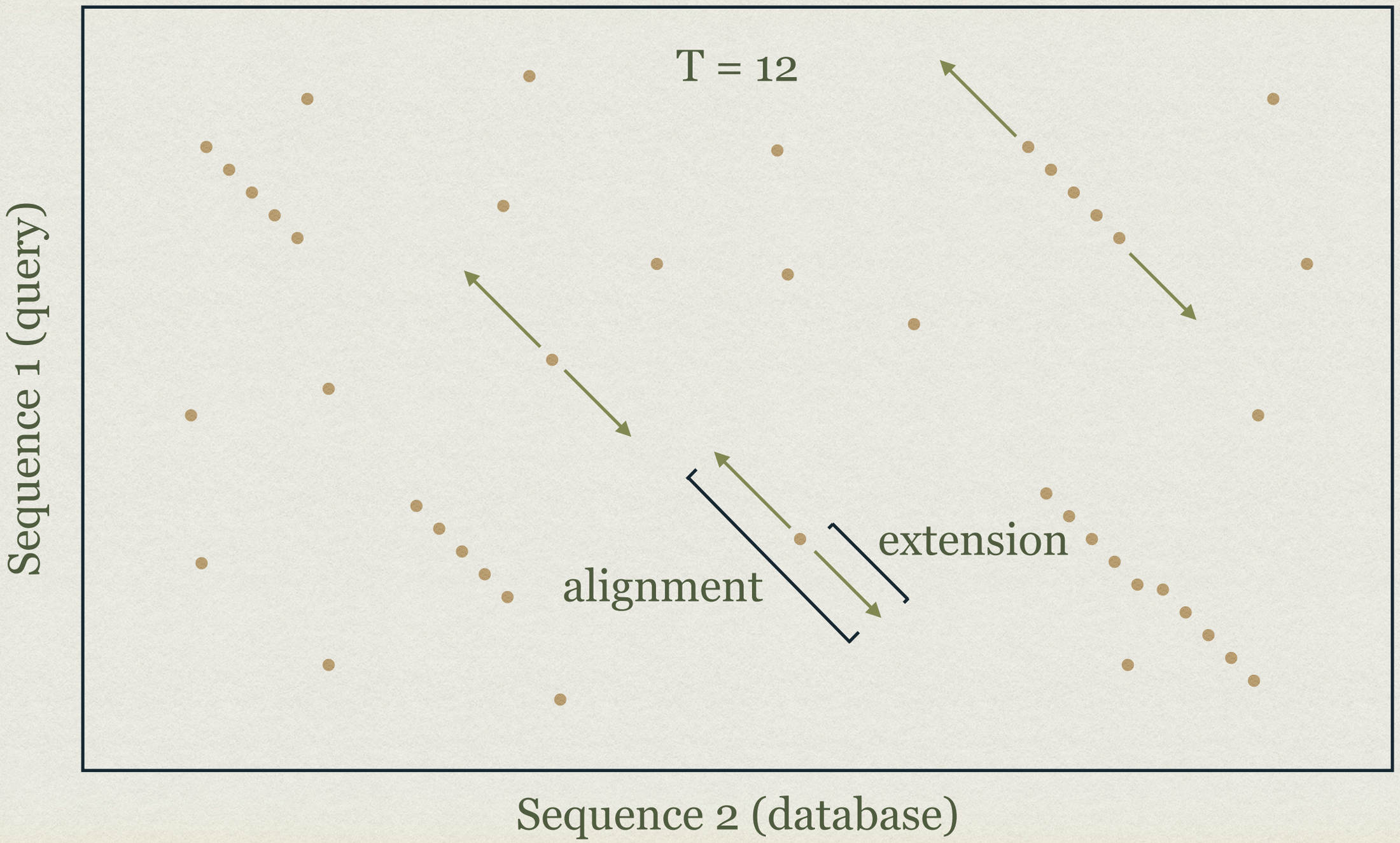
BLAST ALGORITM - SEARCH SPACE



NUCLEOTIDE BLAST ALGORITHM

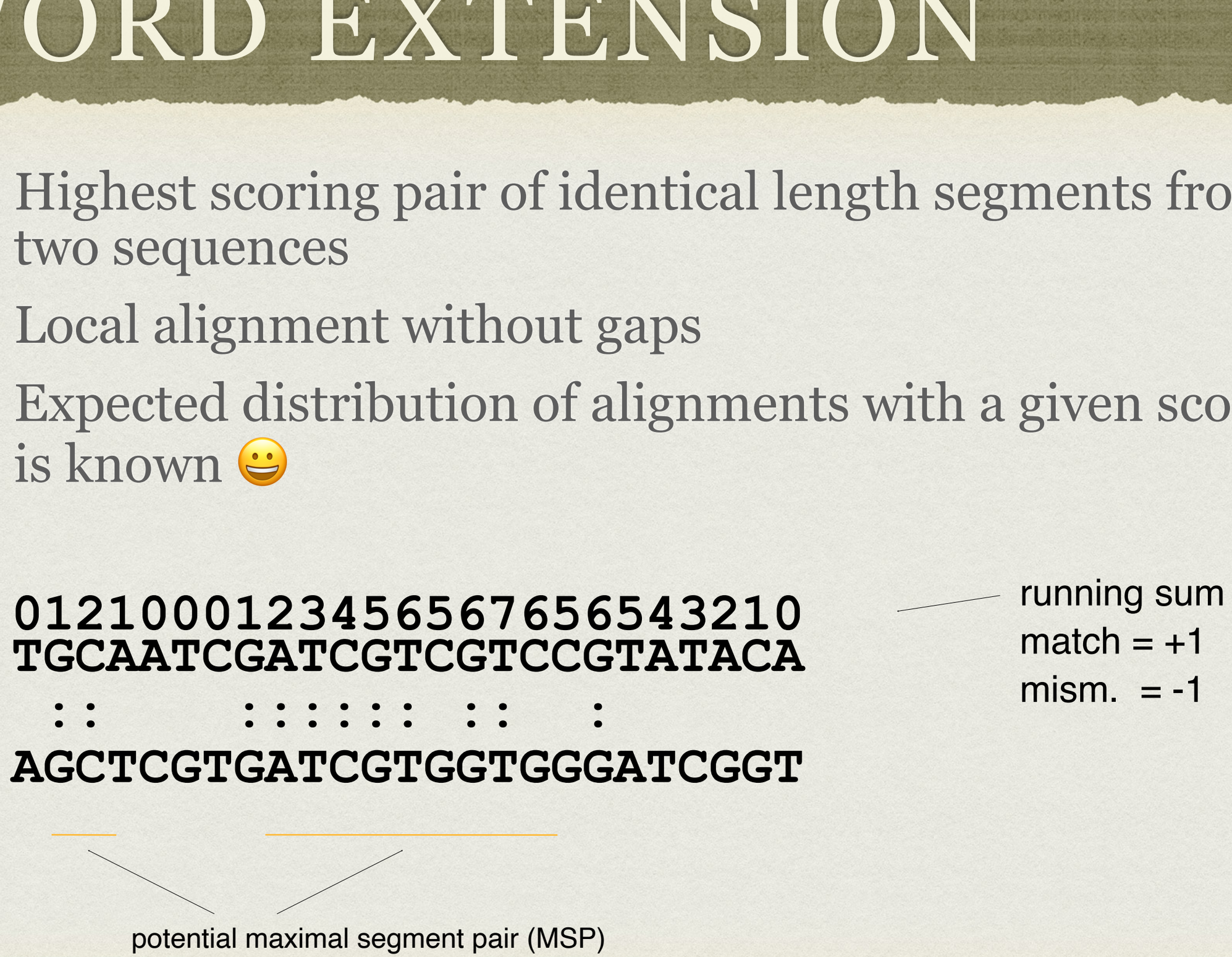
1. Break down query sequence into overlapping words.
2. Scan databases for exact matches of size W (BLASTn) or 110110 pattern (MegaBlast).
3. Try to extend the word matches into the complete maximal scoring pair (MSP). Significance is easily calculated from Karlin-Altschul equation.

BLAST ALGORITM - WORD EXTENSION



BLAST ALGORITM - WORD EXTENSION

- Highest scoring pair of identical length segments from two sequences
- Local alignment without gaps
- Expected distribution of alignments with a given score is known 😀



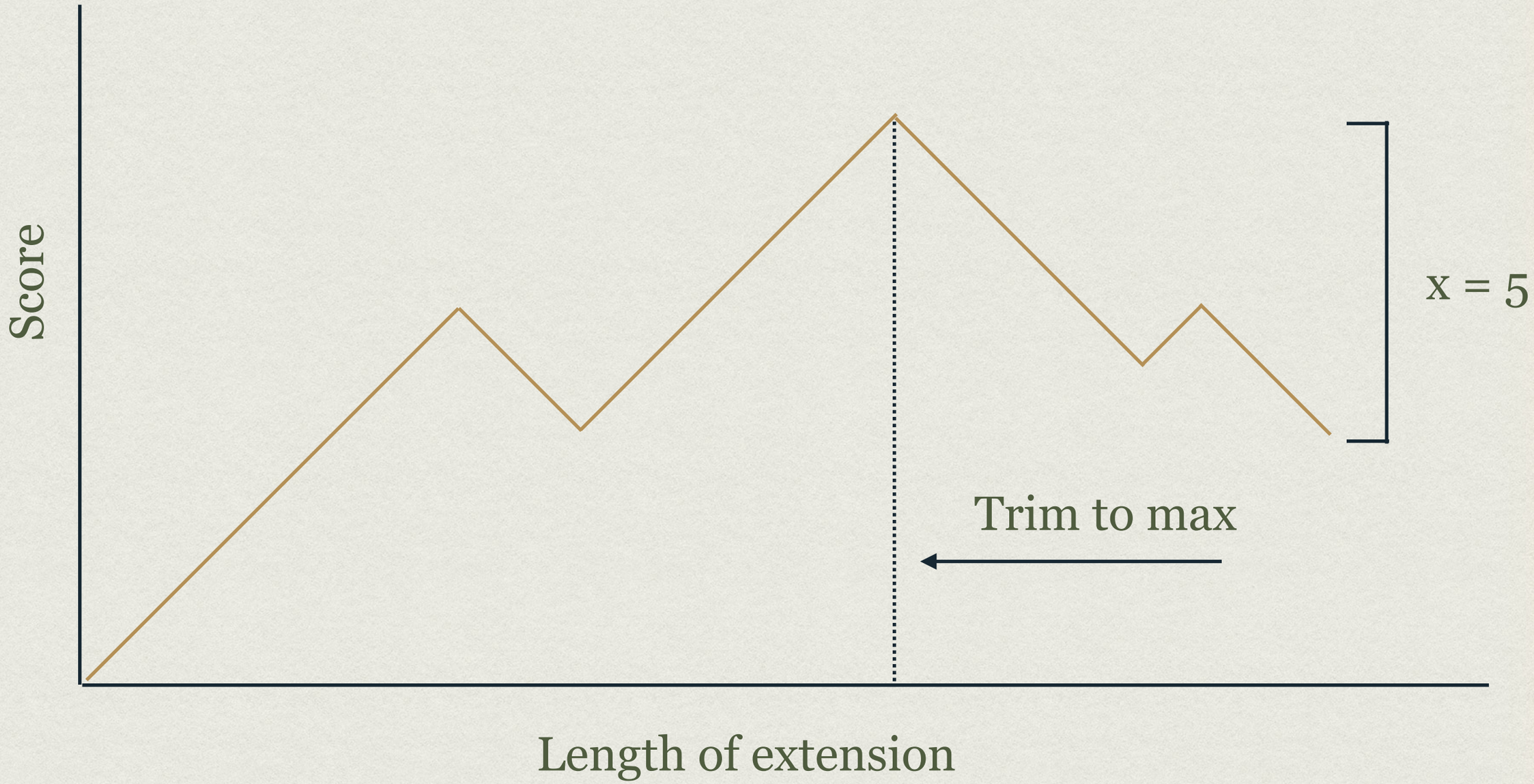
BLAST ALGORITM - WORD EXTENSION

Most expensive step in BLAST algorithm

Extend to end of high scoring segment pair, or HSP. HSPs approximate maximal segment pairs or MSPs. They are only approximate because extension does not continue until running score reaches zero - drop off value concept.

After initial hit was found BLAST tries so called extension - an alignment is extended until the maximum value of the score drops by score x, hence name x dropoff value

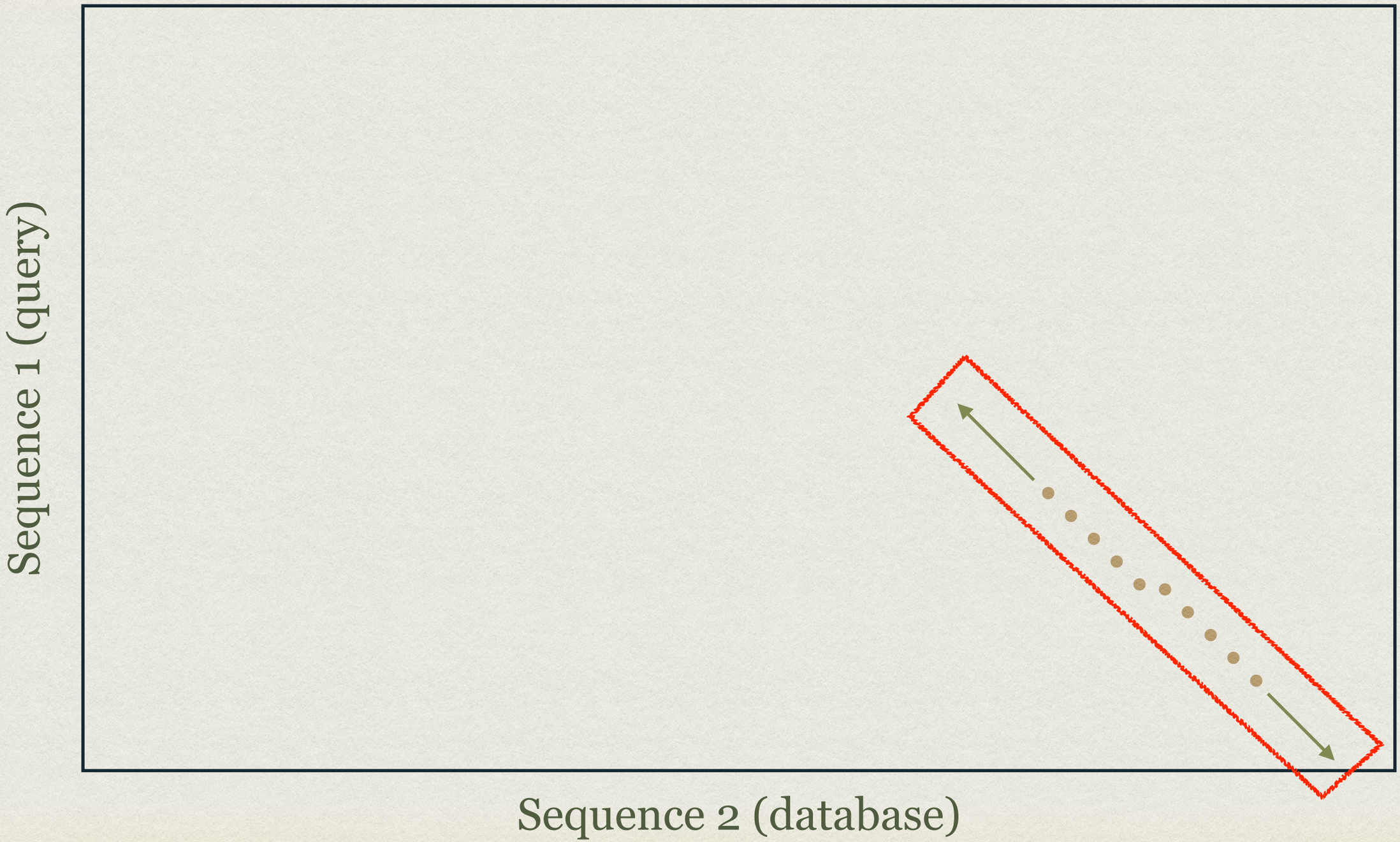
BLAST ALGORITM - WORD EXTENSION



NUCLEOTIDE BLAST ALGORITHM

1. Break down query sequence into overlapping words.
2. Scan databases for exact matches of size W (BLASTn) or 110110 pattern (MegaBlast).
3. Try to extend the word matches into the complete maximal scoring pair (MSP). Significance is easily calculated from Karlin-Altschul equation.
4. Perform local dynamic programming alignment around MSP regions

BLAST ALGORITM - SEARCH SPACE



PROTEIN BLAST ALGORITHM

- Break down query sequence into overlapping words and create a lookaup table.
- For each word, determine a neighborhood of words that, if found in another sequence, would likely to be part of a significant maximum scoring pair (MSP).
- Scan databases for neighborhood words.
- If two words are found on the same diagonal within a specified distance, try to extend the word matches into the complete MSP. Significance is (relatively) easy calculated from Karlin-Altschul equation.
- Perform local dynamic programming alignment around MSP regions
- first step of BLASTp is controlled by three parameters and a score matrix
- w - word length; default value is 3 (lowest possible is 2); two words on the same diagonal are required
- f - score threshold; overall score of the "mini-alignment" has to be above the threshold - the concept of "neighborhood words"

BLASTp - neighborhood words

Example - ITV triplet

	BLOSUM62	PAM230
ITV - ITV	4+5+4 = 13	5+3+5 = 13
ITV - MTV	1+5+4 = 10	2+3+5 = 10
ITV - ISV	4+1+4 = 9	2+3+5 = 10
ITV - LTV	2+5+4 = 11	2+3+5 = 10
ITV - LSV	2+1+4 = 7	2+3+5 = 10
ITV - MSV	1+1+4 = 6	2+3+5 = 10
ITV - IAV	4+0+4 = 8	5+1+5 = 11
ITV - MAV	1+0+4 = 5	2+1+5 = 8
ITV - ITL	4+5+1 = 10	5+3+2 = 10
ITV - LAV	2+0+4 = 6	2+1+5 = 8

BLASTp - neighborhood words

Threshold f = 11 (default for BLASTp)

	BLOSUM62	PAM230
ITV - ITV	4+5+4 = 13	5+3+5 = 13
ITV - MTV	1+5+4 = 10	2+3+5 = 10
ITV - ISV	4+1+4 = 9	2+3+5 = 10
ITV - LTV	2+5+4 = 11	2+3+5 = 10
ITV - LSV	2+1+4 = 7	2+3+5 = 10
ITV - MSV	1+1+4 = 6	2+3+5 = 10
ITV - IAV	4+0+4 = 8	5+1+5 = 11
ITV - MAV	1+0+4 = 5	2+1+5 = 8
ITV - ITL	4+5+1 = 10	5+3+2 = 10
ITV - LAV	2+0+4 = 6	2+1+5 = 8

f=10

	BLOSUM62	PAM230
ITV - ITV	4+5+4 = 13	5+3+5 = 13
ITV - MTV	1+5+4 = 10	2+3+5 = 10
ITV - ISV	4+1+4 = 9	2+3+5 = 10
ITV - LTV	2+5+4 = 11	2+3+5 = 10
ITV - LSV	2+1+4 = 7	2+3+5 = 10
ITV - MSV	1+1+4 = 6	2+3+5 = 10
ITV - IAV	4+0+4 = 8	5+1+5 = 11
ITV - MAV	1+0+4 = 5	2+1+5 = 8
ITV - ITL	4+5+1 = 10	5+3+2 = 10
ITV - LAV	2+0+4 = 6	2+1+5 = 8

Pairs marked in blue would initiate an alignment extension

BLAST - FINAL STEP

- Smith-Waterman algorithm (local dynamic programming), discussed before but limited to regions that include the HSPs
- Significance of alignment with gaps can be evaluated using K and λ estimated from alignments of random sequences with same gap penalty and scoring parameters
- In spite of claims of being “mathematically rigorous” these parameters can only be estimated empirically

KARLIN-ALTCHUL STATISTICS

High scores of local alignments between two random sequences follow Extreme Value Distribution

KARLIN-ALTCHUL STATISTICS

For ungapped alignments their expected number with score S or greater equals

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

K i λ, are parameters related to a search space and scoring system, and m, n represent a query and database length, respectively.

Score can be transformed to a bit-score according to formula **$S' = bitscore = (\lambda S - lnK)/ln2$,** then

$$E = mn2^{-S'}$$

KARLIN-ALTCHUL STATISTICS

- for ungapped alignments parameters K and λ are calculated algebraically but for gapped alignment a solid theory doesn't exist and these parameters are calculated by simulation which has to be run for every combination of scoring system including gap penalties
- therefore not all gap opening and extension score combinations are available
- more at http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

BLAST - KNOWN PROBLEMS

- Significance is calculated versus theoretic distribution using Karlin-Altschul equation not real sequences.
- Assumes sequences are random
- Assume database is one long sequence – length effects are not corrected for
- Statistics are very inaccurate for short queries (ca. 20 characters).
- Be careful when you change BLAST parameters, some of them should be coordinated, e.g. match/mismatch penalty and X-drop off value
- nucleotide BLAST - default parameters tuned up for speed not sensitivity

BLAST ALGORITHM IMPLEMENTATION

Program	Query	Database type
blastn	nt	nt
megablast	nt	nt
blastp	aa	aa
blastx	nt	aa
tblastn	aa	nt
tblastx	nt	nt
blast2seq	nt, aa	nt, aa

BIOINFORMATICS CREED

Remember about biology

Do not trust the data

Use comparative approach

Use statistics

Know the limits

Remember about biology!!!