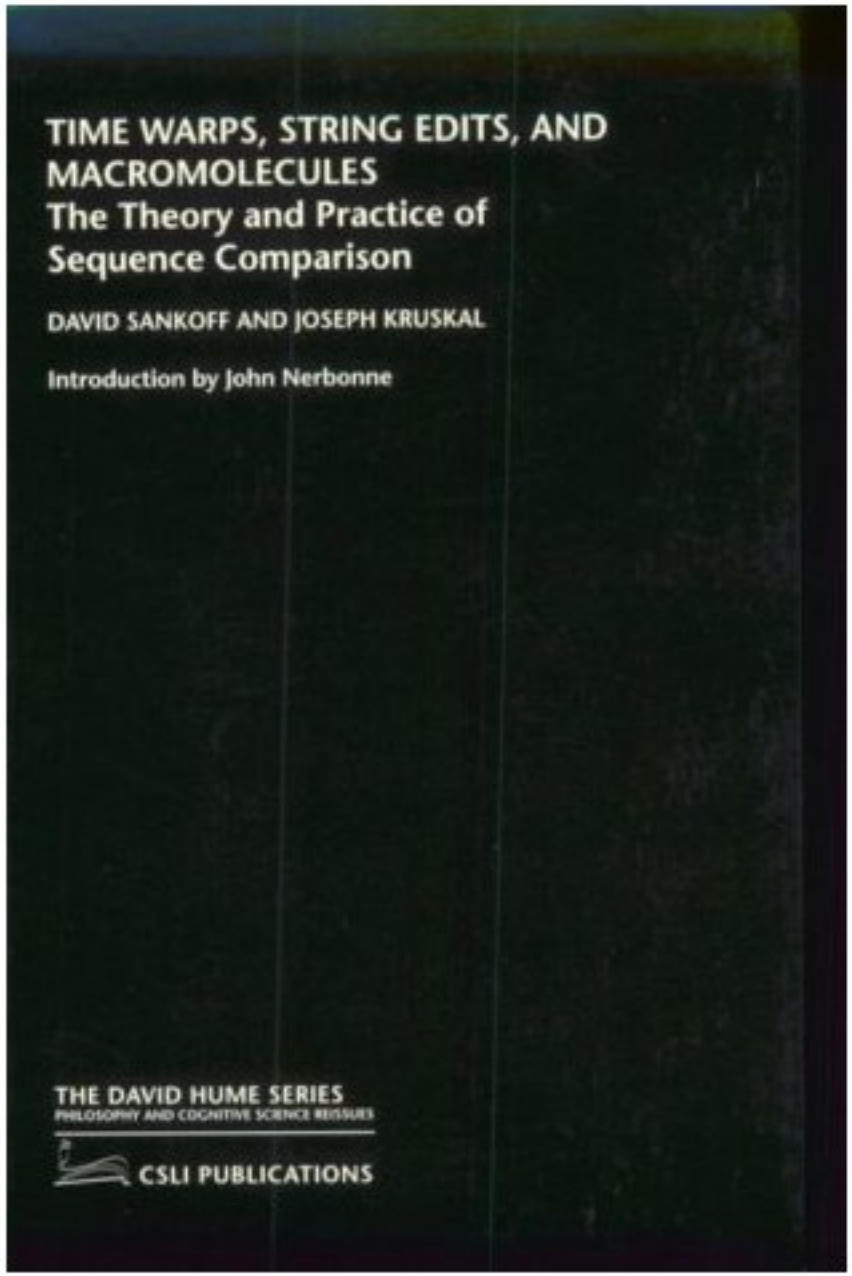
Course Goals:

(1) Introduce statistical methods important for DNA and protein sequence analysis. Emphasis is on getting students to implement these methods.

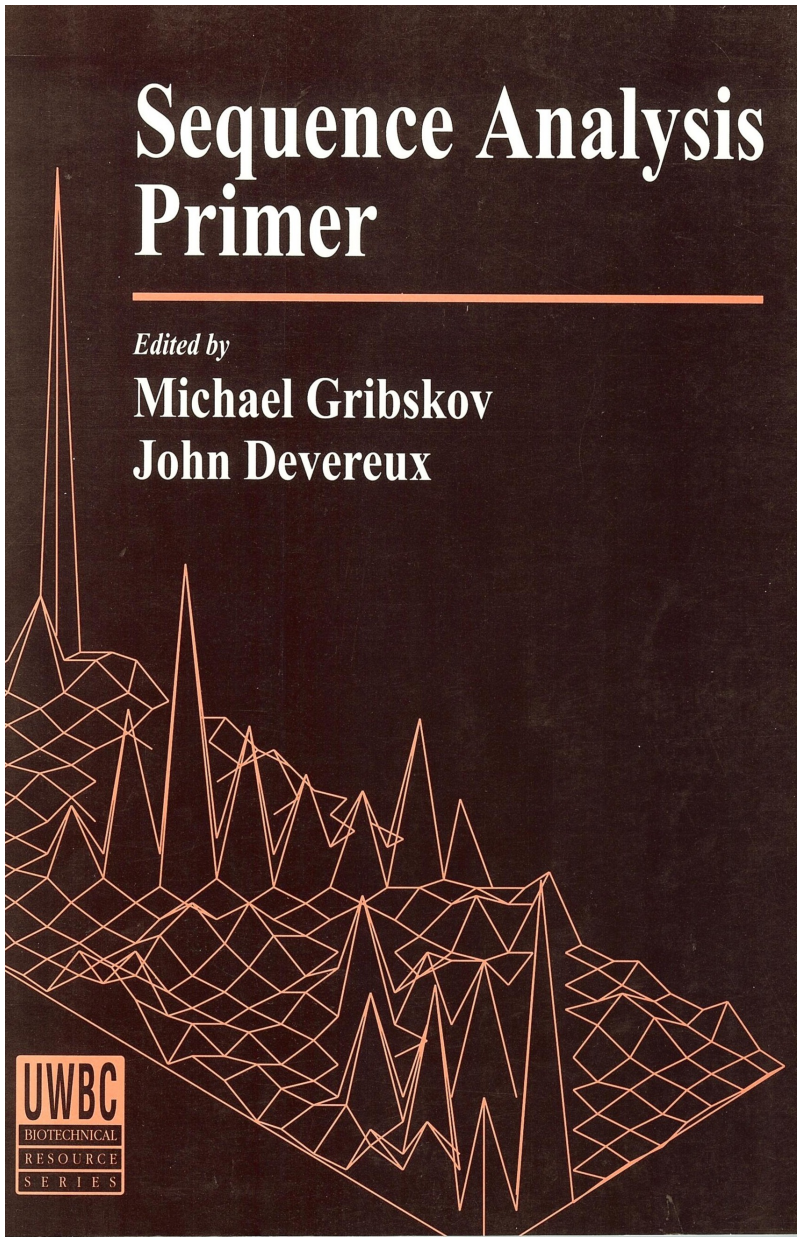
(2) Other (e.g., discuss protein tertiary structure prediction, RNA secondary structure prediction, important sequence analysis algorithms, ...)

Two early and influential books ...



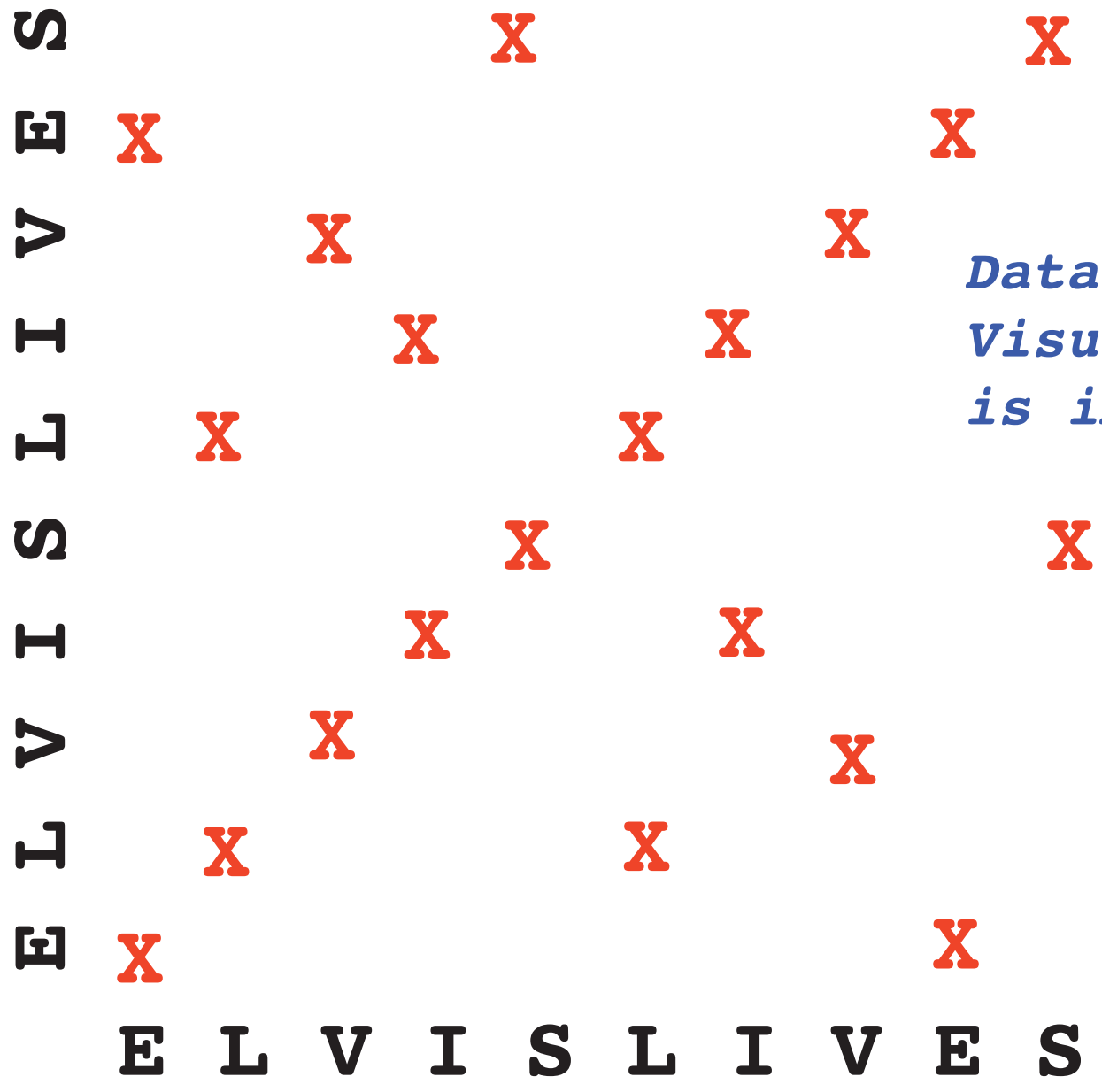
1983

from http://ecx.images-amazon.com/images/I/41JJ612NWVL._SS500_.jpg



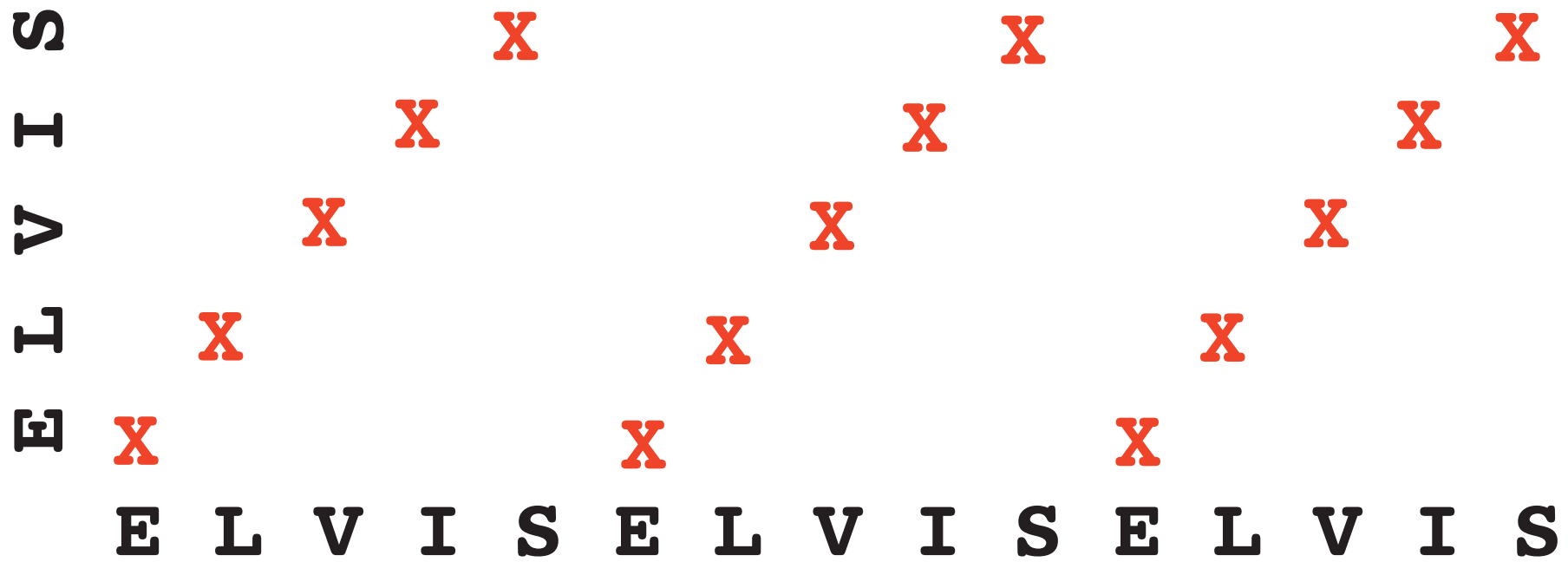
1991

from http://markboguski.net/images/Sequence-Analysis-Primer.jpg

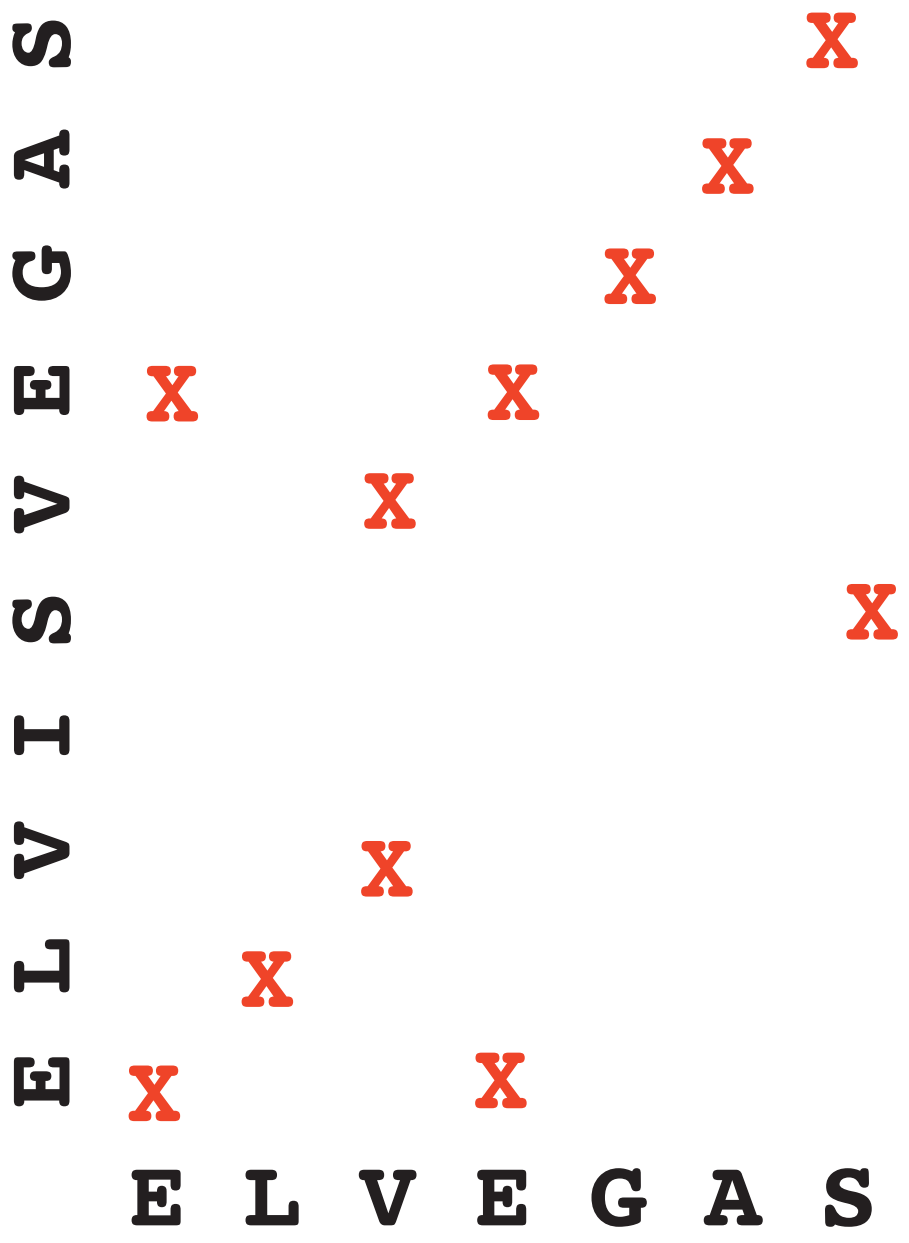


Data Visualization is important!

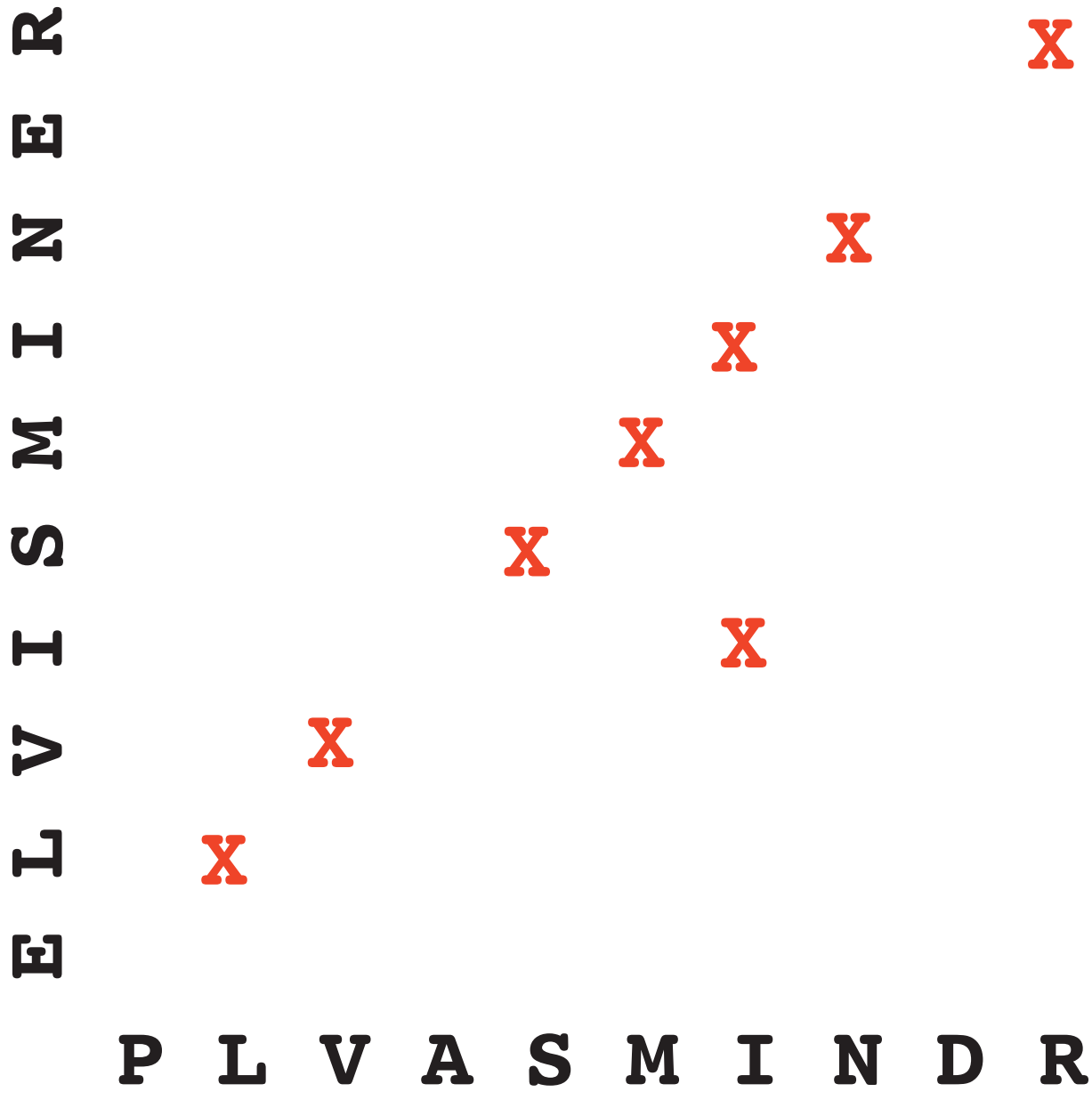
Dot Plot of a sequence versus itself (notice the diagonal pattern)



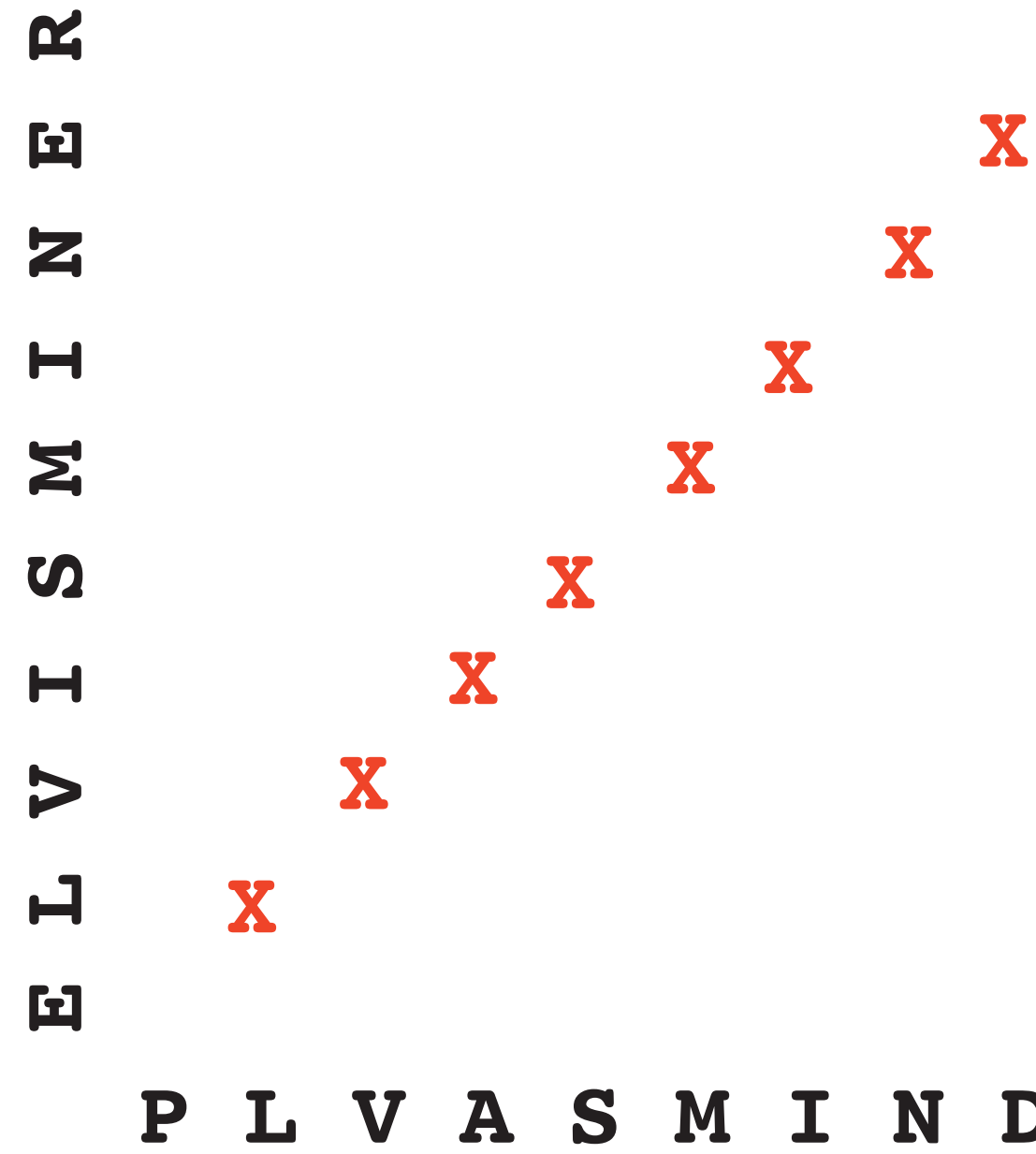
Dot Plots can show repeats



Dot Plots can indicate insertions or deletions (i.e., “indels”)

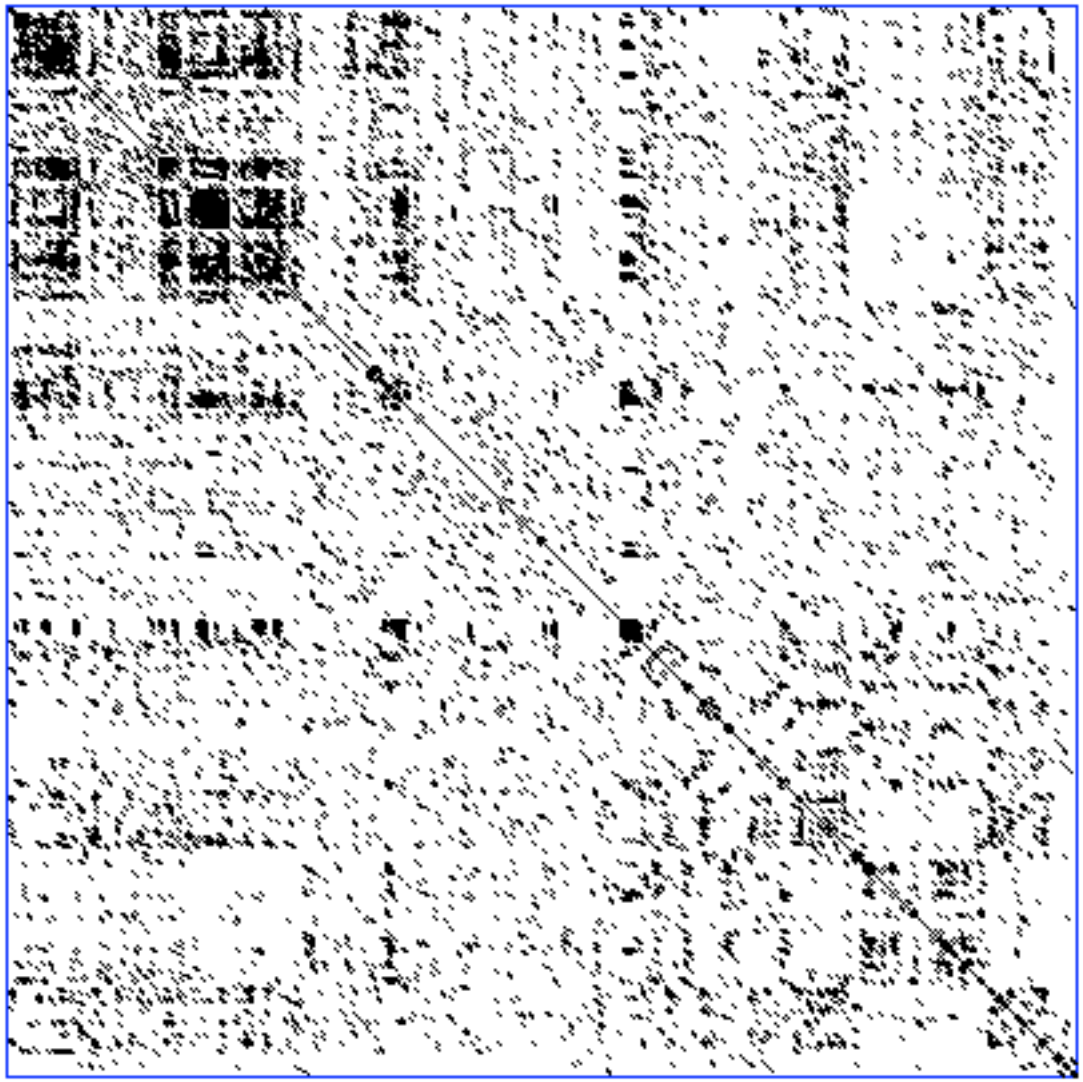


Dot Plots can indicate sequence divergence



Filtering can reduce noise and help with visualizing sequence similarity despite some divergence

Here, a dot is shown only when at least 2 amino acids are identical in window of size 3 that is centered on row and column



An old and good coverage
of dot plots can be found in
“Sequence Analysis Primer”,
Edited by Gribskov and Devereux

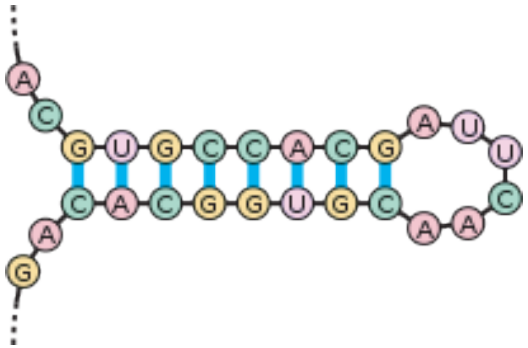
“A DNA dot plot of a human zinc finger transcription factor (GenBank ID NM_002383), showing regional self-similarity. The main diagonal represents the sequence's alignment with itself; lines off the main diagonal represent similar or repetitive patterns within the sequence.”

Copied from http://en.wikipedia.org/wiki/Dot_plot_(bioinformatics)

DOT PLOT VARIANT:

A & U are complementary
G & C are complementary

Black marks signify 3 matches in a row between sequence and reverse complement



Sequence --->

A C G U G G C C A C G A U U C A A C G U G G C A C A G

C U G U G C C A C G U U G A A U C G U G G C A C G U

Reverse Complement --->

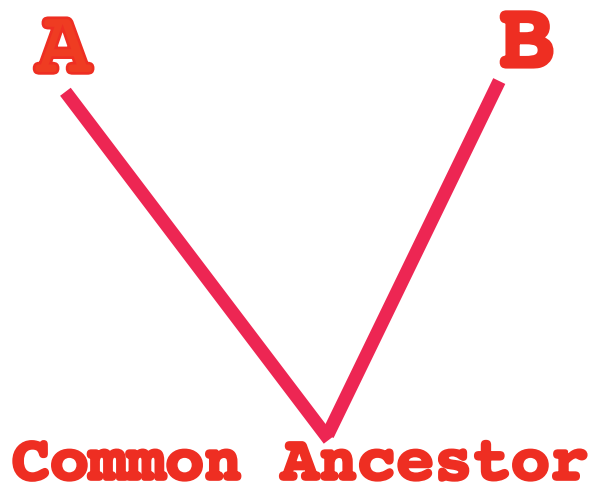
stem-loop image from:
https://en.wikipedia.org/wiki/Nucleic_acid_secondary_structure#/media/File:Stem-loop.svg

Alignments are designed to exhibit evolutionary correspondence between sequences

Typically, they contain matches, mismatches, and gaps

INDEL - INsertion or DELetion

```
A:    C C T - - A T G C T
B:    C C G G G A T G C A
```



Local Alignment - hypothesis about evolutionary correspondence between sequence regions

Global Alignment - hypothesis about evolutionary correspondence along entire sequences

Uses of Alignments

- Detecting relationships between sequences (database searches)
- Pinpointing conserved regions, defining functional motifs
- Prediction of protein structure
- Evolutionary biology, both within and between population comparisons

The basic and widely-used dynamic programming algorithm ...

Original Paper: Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins. J Mol Biol 48:444-453

Similarity-based methods find alignment with maximum associated score

Distance-based methods find alignment with minimum associated penalty (distance)

For a distance-based method, each postulated evolutionary event has an associated penalty

Penalty of Alignment = (Penalties associated with matches) +
(Penalties associated with mismatches) +
(Penalties associated with gaps)

Pairwise Alignment = alignment of 2 sequences

Multiple Alignment = alignment of >2 sequences

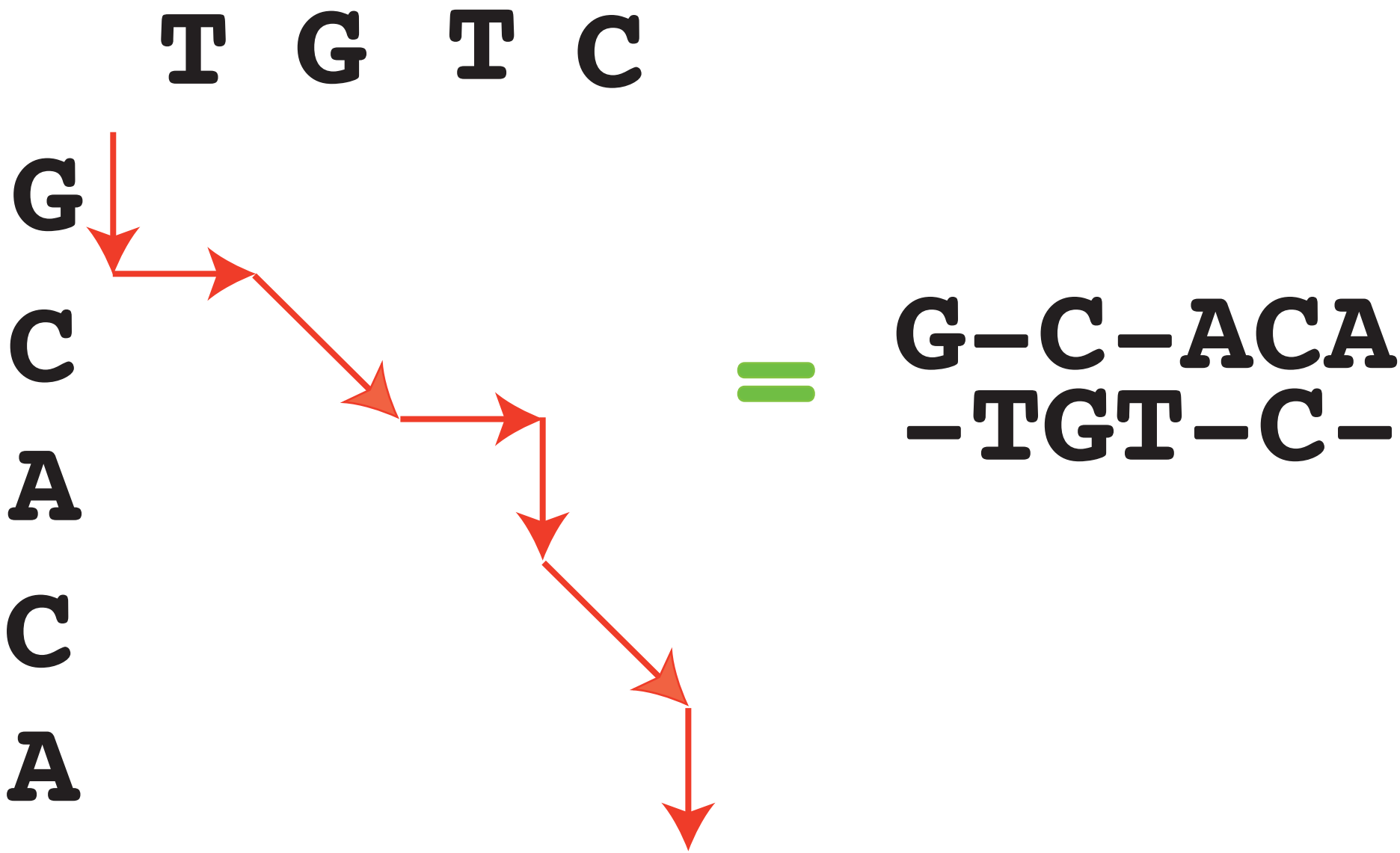
The very basic distance-based dynamic programming algorithm (explanation lifted from Chapter 1 of book edited by Sankoff & Kruskal)

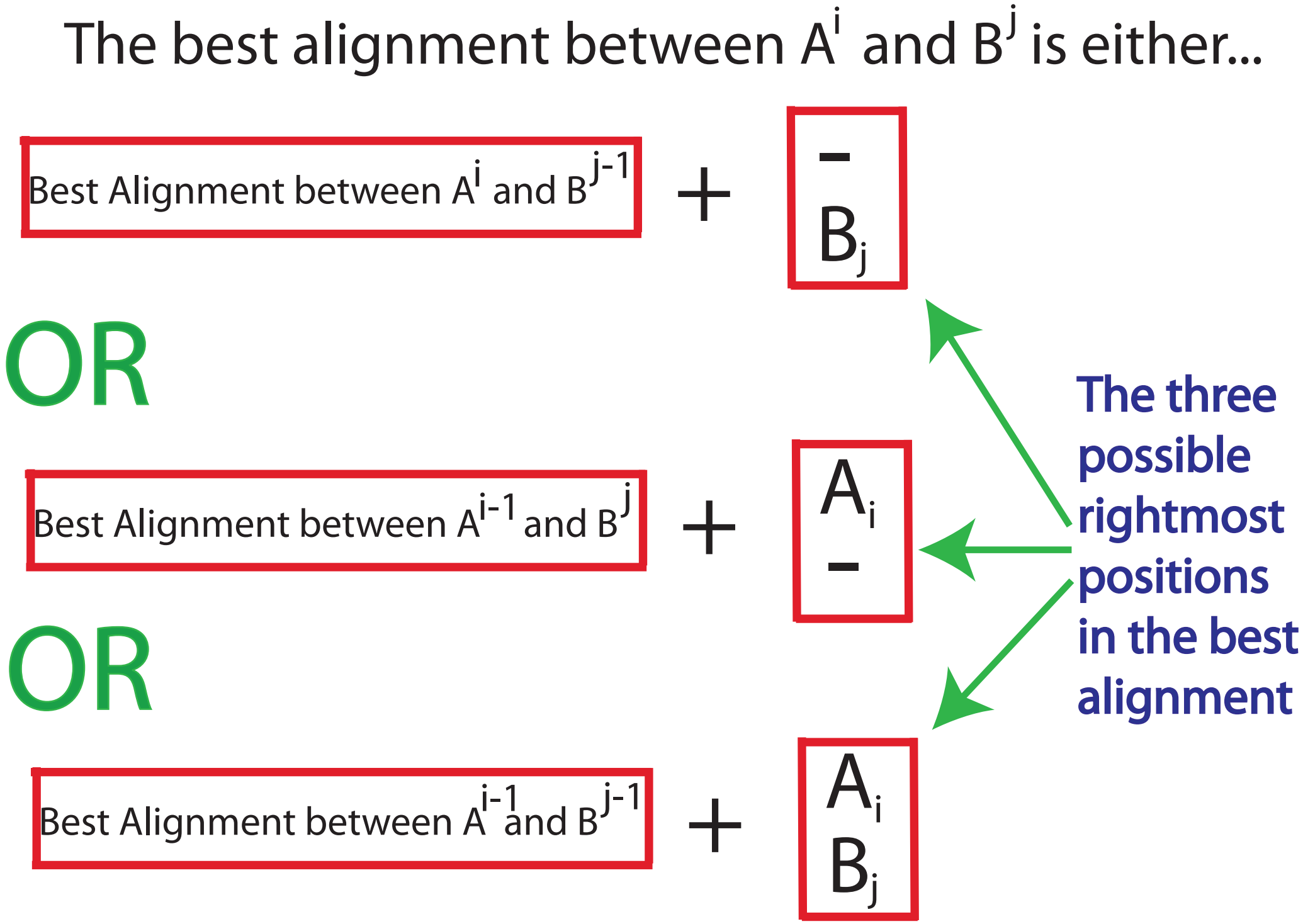
Notation: Let A_i and B_j respectively be i^{th} base of sequence A and j^{th} base of sequence B.

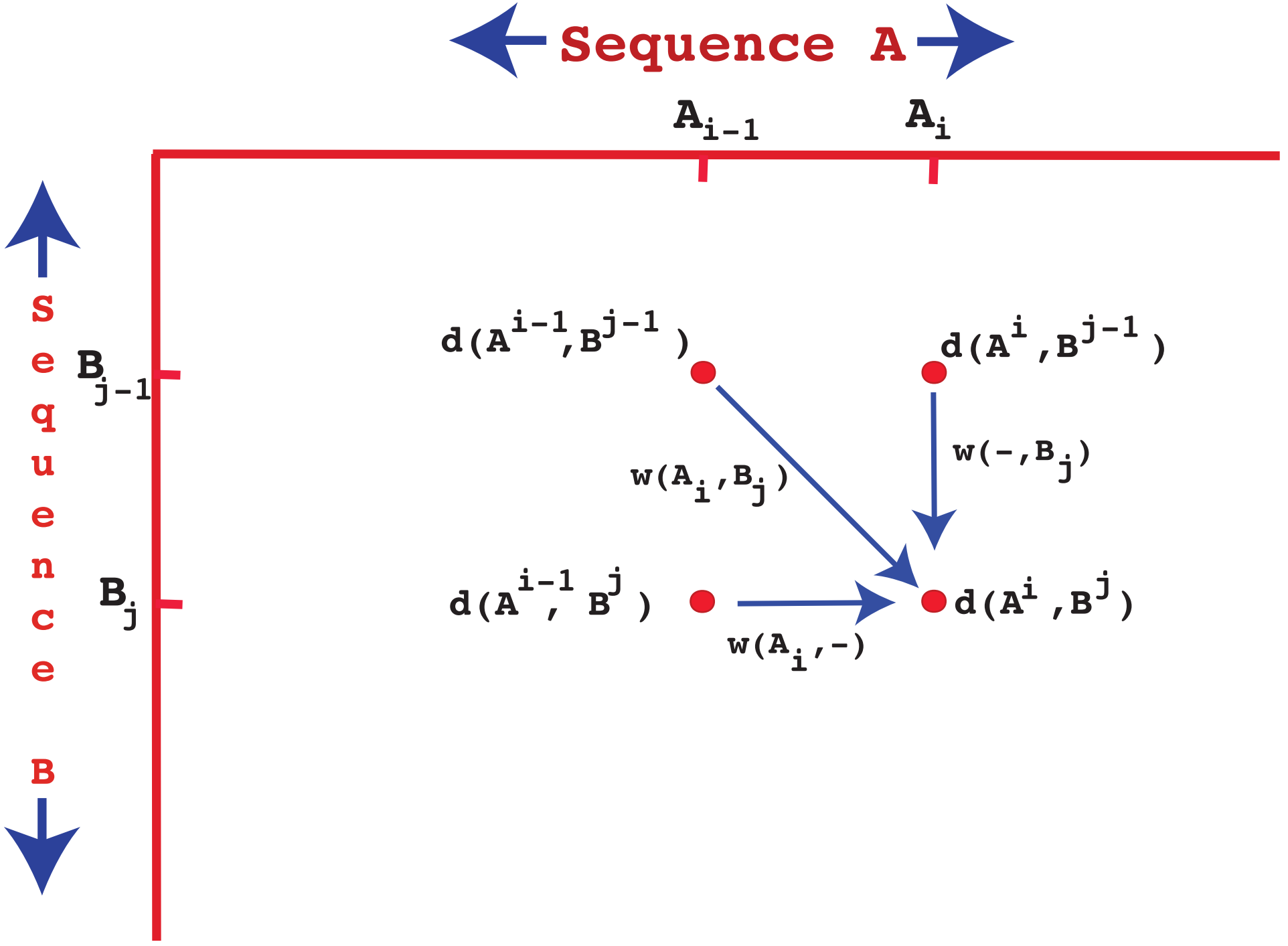
Let A^i and B^j respectively be first i bases of sequence A and first j bases of sequence B.

Let $d(A^i, B^j)$ be the penalty (distance) associated with best alignment between A^i and B^j

Let w(i,j) be penalty for aligning type i with type j (for DNA, i and j will be either A,C,G,T, or "-")


T G T C
G
C
A
C
A
=
G-C-ACA
-TGT-C-


The best alignment between A^i and B^j is either...
Best Alignment between A^i and B^{j-1}
+
-
B_j
OR
Best Alignment between A^{i-1} and B^j
+
A_i
-
OR
Best Alignment between A^{i-1} and B^{j-1}
+
A_i
B_j
The three possible rightmost positions in the best alignment


Sequence A
A_{i-1}
A_i
Sequence B
B_{j-1}
B_j
$d(A^{i-1},B^{j-1})$
$d(A^i,B^{j-1})$
$w(A_i,B_j)$
$w(-,B_j)$
$d(A^{i-1},B^j)$
$d(A^i,B^j)$
$w(A_i,-)$

Notation: G_k = Penalty for gap of length k
= k times penalty for gap of length 1

Initial Condition: $d(A^0, B^0) = 0$

$$d(A^i, B^j) = min \begin{cases} d(A^{i-1}, B^j) + w(A_i, -) \\ d(A^{i-1}, B^{j-1}) + w(A_i, B_j) \\ d(A^i, B^{j-1}) + w(-, B_j) \end{cases}$$

To recover best alignment (and not just score of best alignment), "traceback" step is required.

$d(A^0, B^0) = 0$

For $i>0$,

$d(A^i, B^0) = ?$

Let g be the penalty of an alignment column with a gap.

For $j>0$,

$d(A^0, B^j) = ?$

$$d(A^0, B^0) = 0$$

Let g be the penalty of an alignment column with a gap.

For $i>0$,

$$d(A^i, B^0) = gi$$

$$d(A^i, B^0) = d(A^{i-1}, B^0) + g$$

For $j>0$,

$$d(A^0, B^j) = gj$$

$$d(A^0, B^j) = d(A^0, B^{j-1}) + g$$

n is size of data set. We think about situation where n is really big!

Amount of computation needed to solve problem:

$an + b$ is linear (i.e., $O(n)$) no matter what value of $a>0$ and $b > 0$

$an^2 + bn + c$ is $O(n^2)$

...

For general polynomial

$an^k + bn^{k-1} + ... + yn^2 + zn + c$ is $O(n^k)$

For big enough n, polynomial running time is better than exponential ...

$e^{hn} + c$

... no matter what is the value of k for the above polynomial

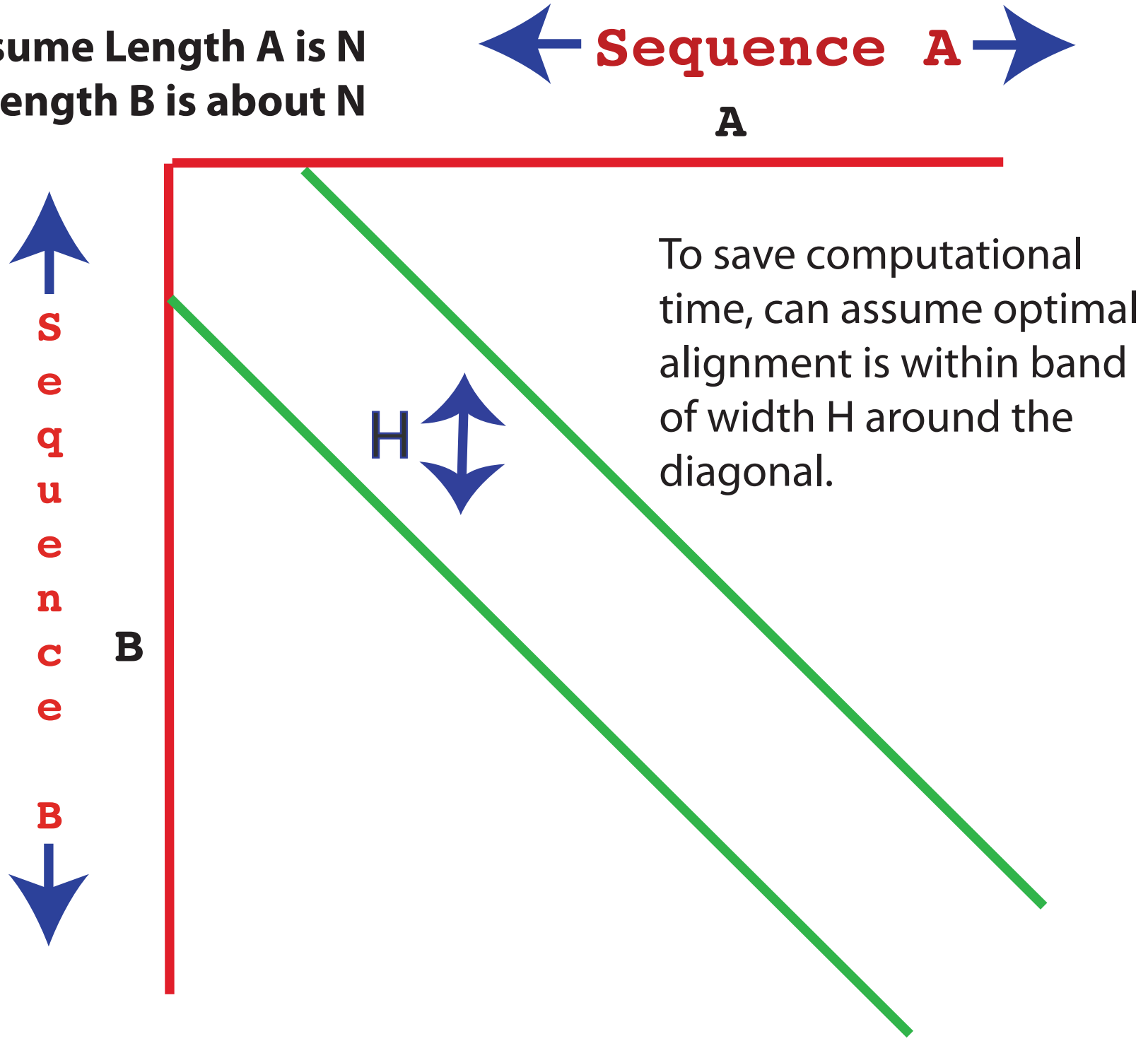
If M and N are the 2 sequence lengths, algorithm with above form of G_k takes amount of computation proportional to MN (also works for more generic concave penalty function).

Also, algorithm as presented requires amount of storage proportional to MN. But, clever modification can reduce storage to be proportional to minimum of M and N at the cost of doubling expected amount of computation ...

Shortcuts to reduce computation are also available. In practice, these almost always help but they are not guaranteed to do so...

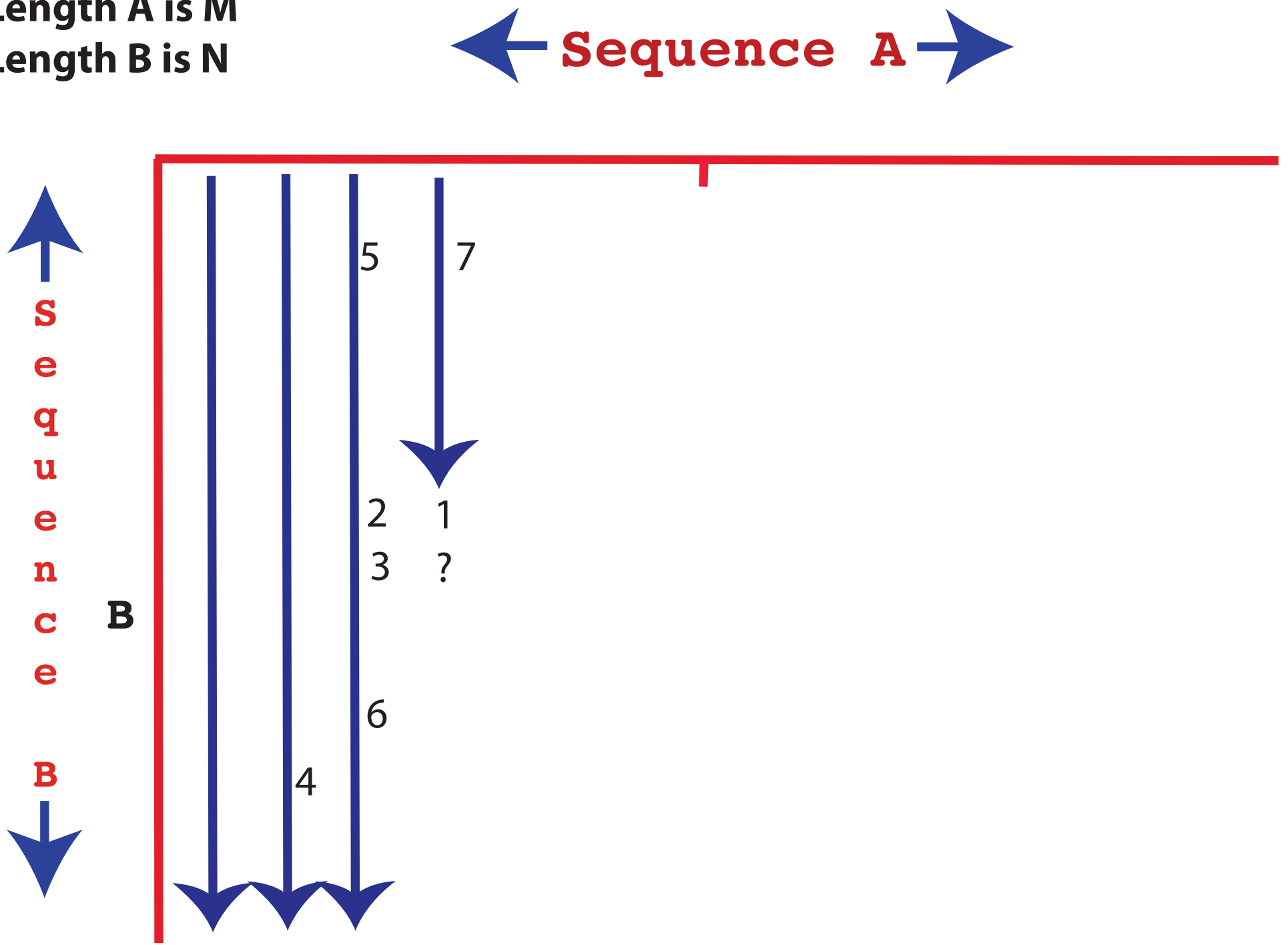
Scaling this algorithm to more than 2 sequences ...

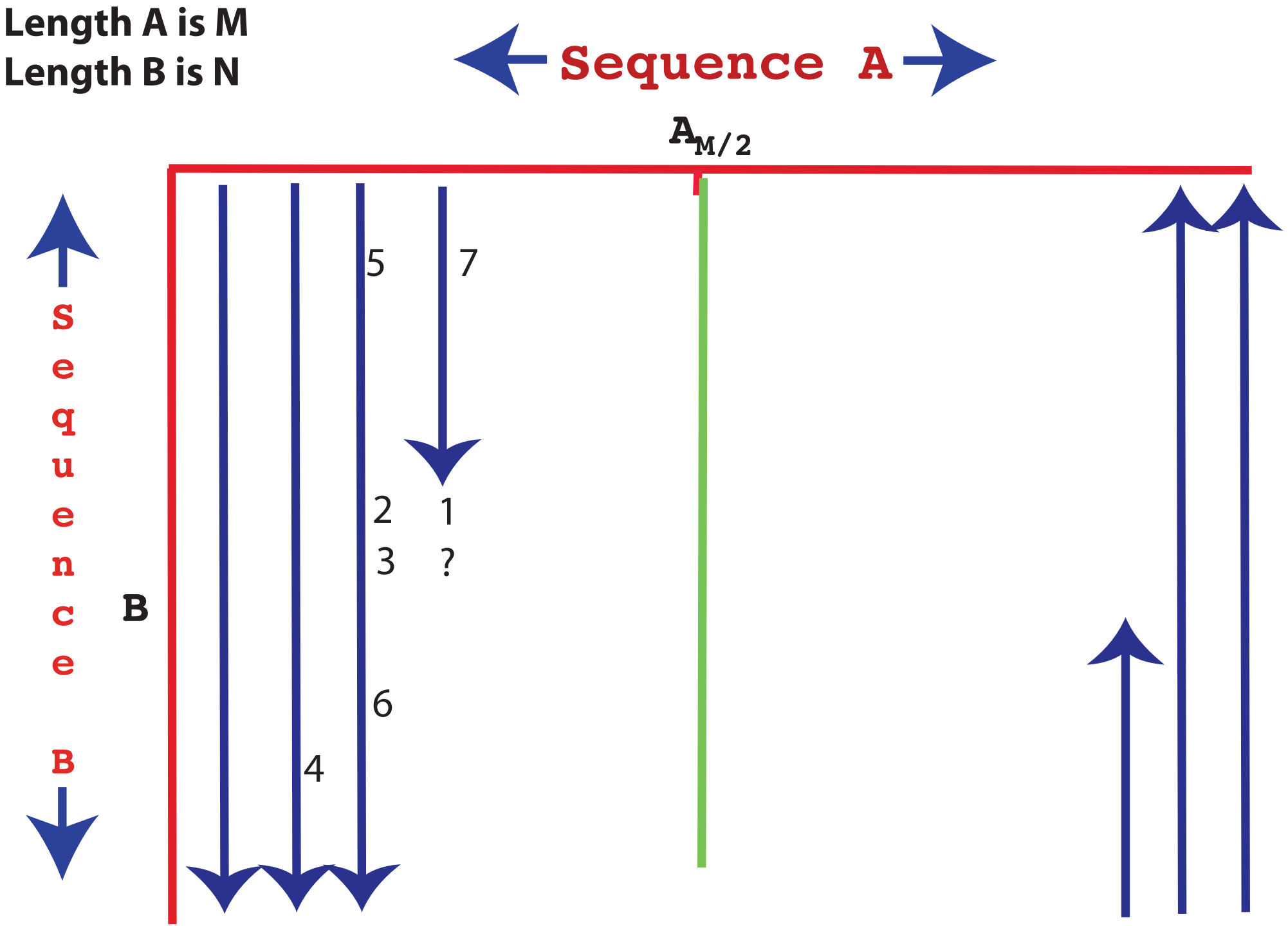
Assume Length A is N
& Length B is about N

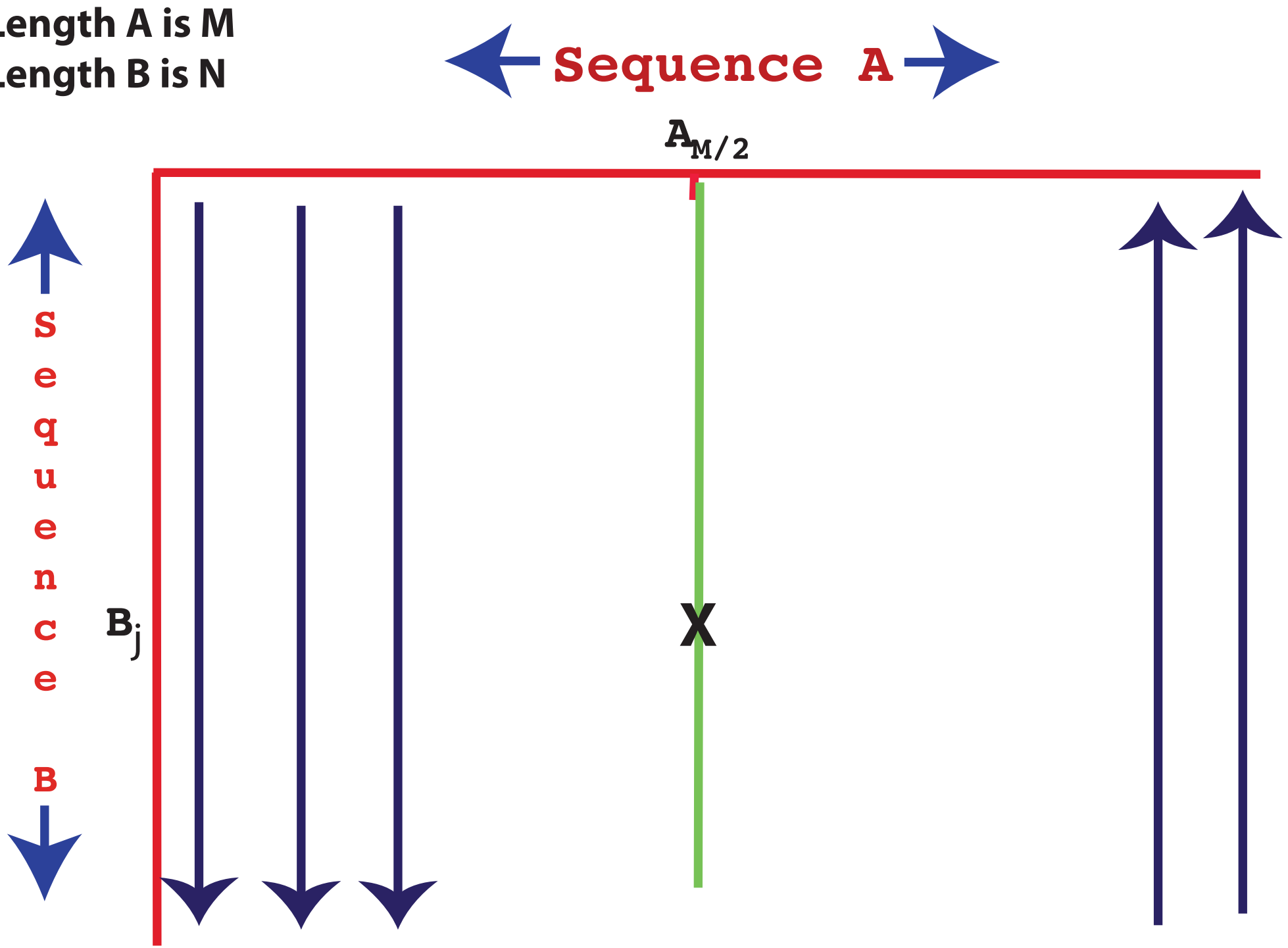


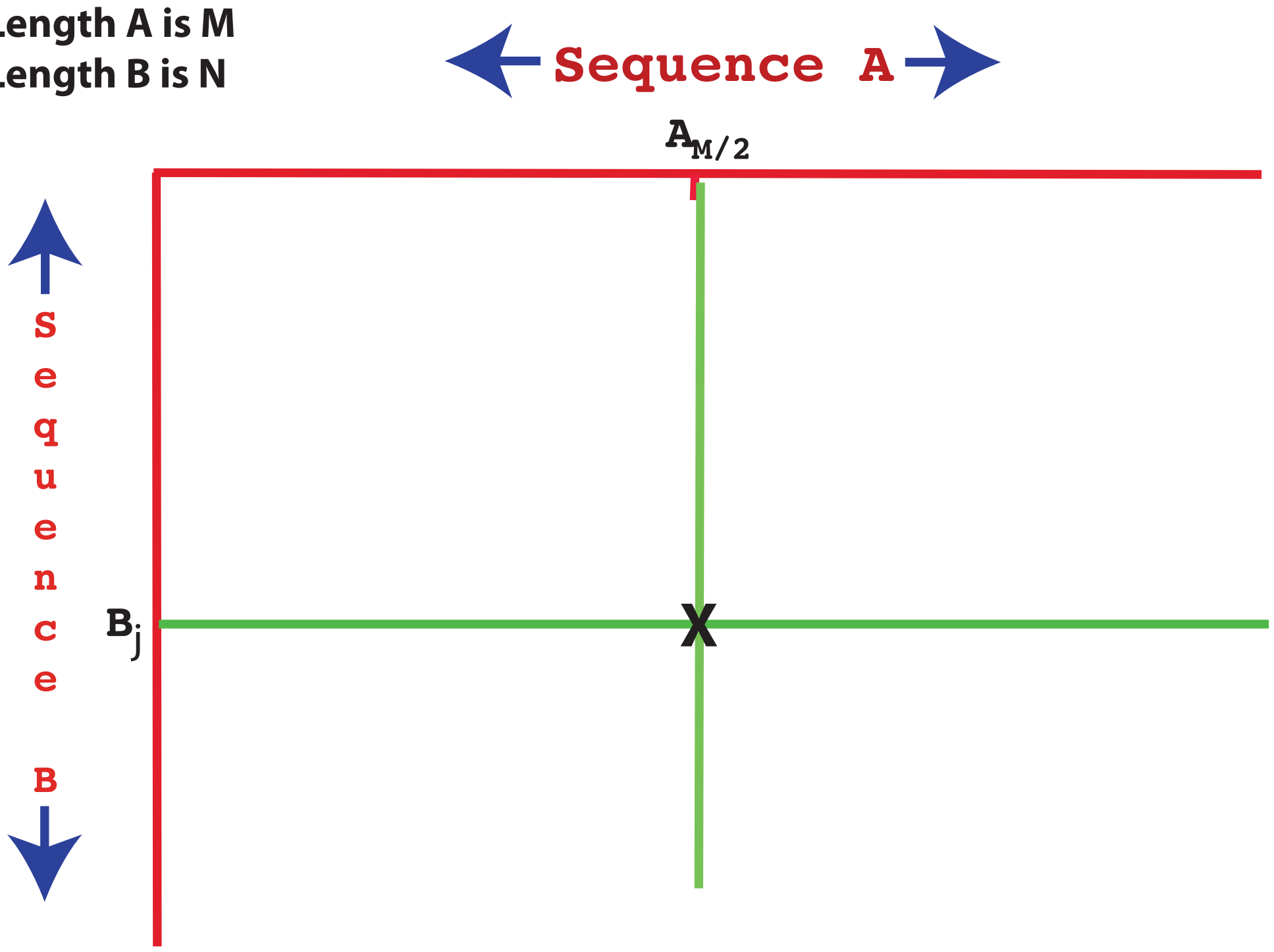
To save computational time, can assume optimal alignment is within band of width H around the diagonal.

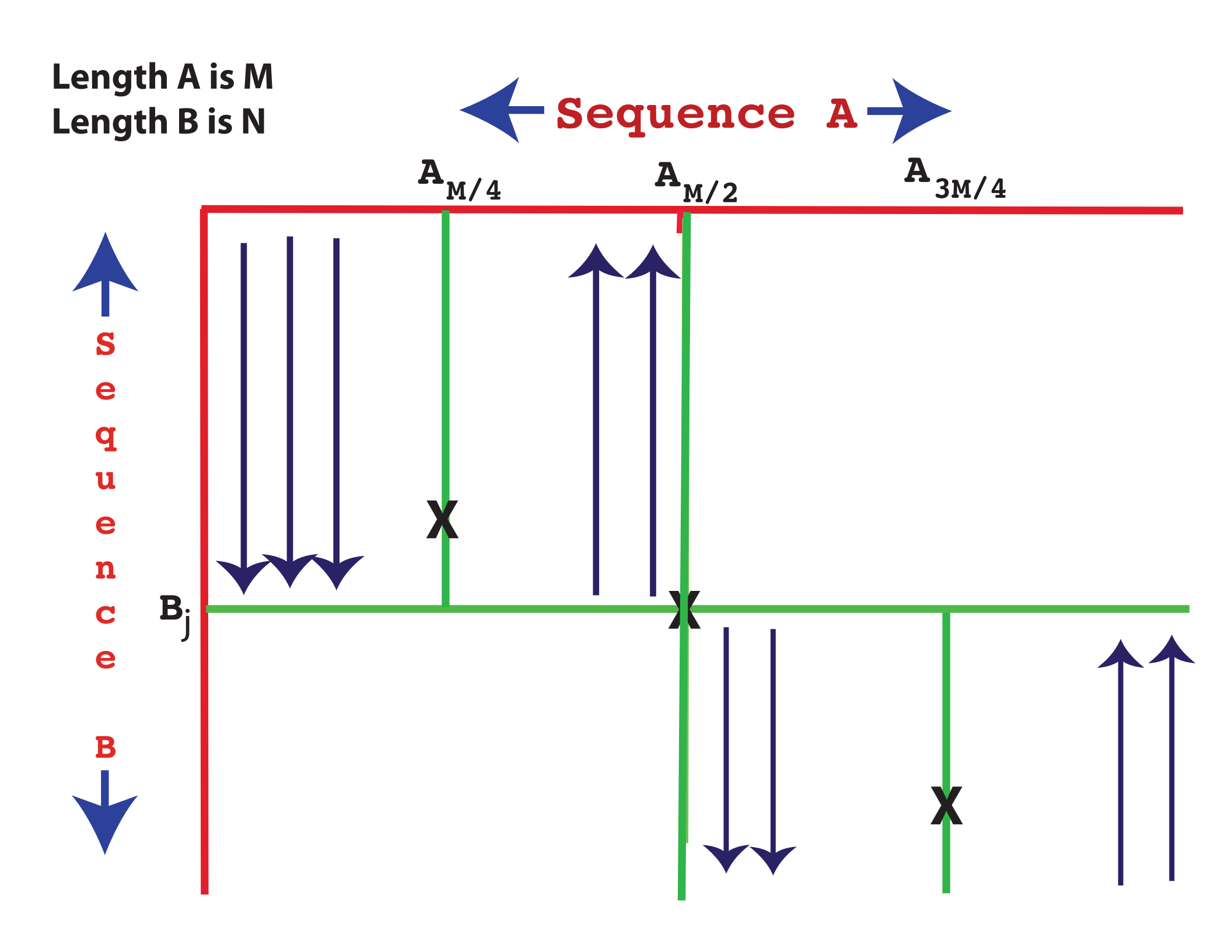
Length A is M
Length B is N

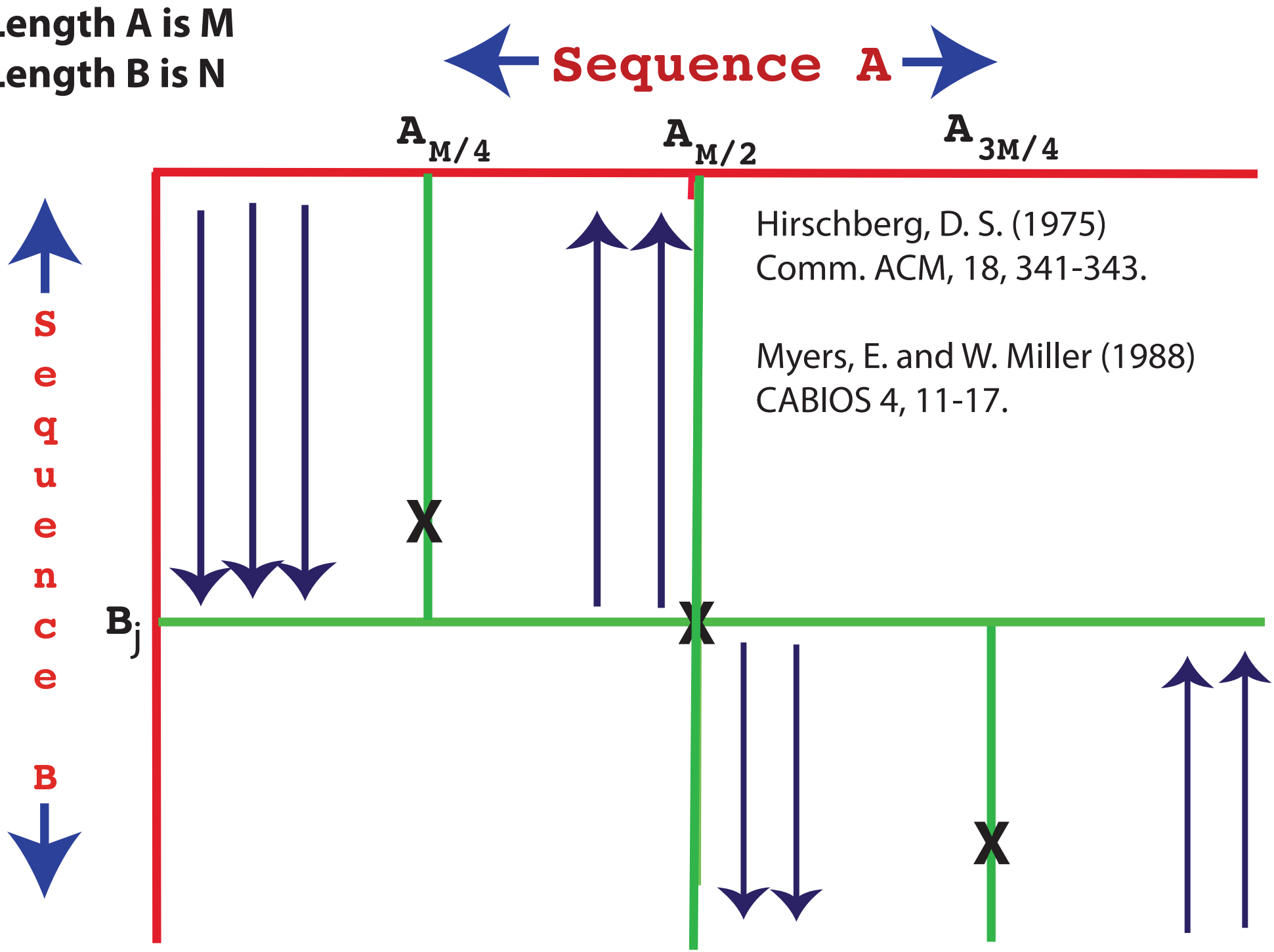



Length A is M
Length B is N
Sequence A
$A_{M/2}$
Sequence B
B
5
7
2
1
3
?
6
4


Length A is M
Length B is N
Sequence A
$A_{M/2}$
Sequence B
B_j
X


Length A is M
Length B is N
Sequence A
$A_{M/2}$
Sequence B
B_j
X


Length A is M
Length B is N
Sequence A
$A_{M/4}$
$A_{M/2}$
$A_{3M/4}$
Sequence B
B_j
X
X
X


Length A is M
Length B is N
Sequence A
$A_{M/4}$
$A_{M/2}$
$A_{3M/4}$
Sequence B
B_j
Hirschberg, D. S. (1975)
Comm. ACM, 18, 341-343.
Myers, E. and W. Miller (1988)
CABIOS 4, 11-17.

Align-ment	Match Weight	Mismatch Weight	Indel Weight
(A)	0	2	3

Best Alignment:
AGTCAG-C
A-TCAGTC

(B)	0	2	10

Best Alignment:
AGTCAGC
ATCAGTC

Message: Different Sequence Pairs require different sets of weights*

Which alignment is least biologically plausible?

```
GCAGAAACGTA
GCAG--ACGTA
```

or

```
GCAGAAACGTA
GCAG-A-CGTA
```

or

```
GCAGAAACGTA
GCAGA--CGTA
```

* More Subtle Message -- A biologically "correct" form of the weighting scheme is not obvious. Commonly, $G_k = a + bk$ where k is the length of a gap of size k. This convention is adopted for algorithmic reasons, not biological reasons.

* More Subtle Message -- A biologically "correct" form of the weighting scheme is not obvious. Commonly, $G_k = a + bk$ where k is the length of a gap of size k. This convention is adopted for algorithmic reasons, not biological reasons.

(Linear Gap penalty for gap of size k) = bk

(Affine Gap penalty for gap of size k) = a + bk
or equivalently ... = (a+b) + b(k-1)

$E_1, E_2, ..., E_N$ are independent events.

$$P(E_1, E_2, ..., E_N) = P(E_1) x P(E_2) x ... x P(E_N)$$

$$\log(P(E_1, E_2, ..., E_N)) = \log(P(E_1)) + \log(P(E_2)) + ... + \log(P(E_N))$$

Message: Alignment score is sum of scores of individual columns

Finding maximum alignment score (i.e., similarity-based scoring) is kind of like maximizing log probability of alignment

Minimizing alignment score (i.e., distance-based scoring) is kind of like minimizing negative log probability of alignment

(Affine Gap penalty for gap of size k) = a + bk

or equivalently ... = (a+b) + b(k-1)

"a+b" is gap opening penalty, "b" is gap continuation penalty

Geometric Distribution defined on positive integers k= 1,2,3, ...

$P(i=k) = (1-r)\, r^{k-1}$ where $0<r<1$

Mean of this distribution is 1/(1-r)

$\log(P(i=k)) = \log(1-r) + \log(r) \times (k-1)$

$-\log(P(i=k)) = -\log(1-r) - \log(r) \times (k-1)$