Define “Alignment”

An alignment of two sequences (frequently called a *local alignment*) can be obtained as follows.

1. extract a segment from each sequence
2. add dashes (gap symbols) to each segment to create equal-length sequences
3. place one padded segment over the other

For example:

```
AACC-GTACTTG
A-CAGGTGG-TG
```

Alignment Scores

We need to differentiate good alignments from poor ones. We use a rule that assigns a numerical score to any alignment; the higher the score, the better the alignment.

For any proposed rule for scoring an alignment, there are two questions:

1. Given any alignment, can we compute its score?
2. Given two sequences, can we automatically find a local alignment of highest possible score?

For some rules, the second answer is “No”.

Simple Rule for Scoring Alignments

We give a score to each possible column, then add scores of an alignment's columns.

Let a match (column with identical symbols) score 1 and each other column score –1. For example:

```
AACC-GTACTTG
A-CAGGTGC-TG
+-+--++-+-++
```

Total score is 2.

Optimal Alignments

With this scoring method, for any two sequences we can compute a highest-scoring local alignment (in time proportional to the product of the two sequence lengths, using “dynamic programming”).

Needleman and Wunsch (1970); Smith and Waterman (1981)

Unusable Rule for Scoring Alignments

Again, each mismatch scores –1. A match column scores $n/(n+1)$, where n is the number of match columns for that same letter (thus the n identical matches total $n^2/(n+1)$).

```
AACC-GTACTTT
A-CAGGTGC-TT
+-+--++-+-++
```

There are 5 mismatch columns (score –5), 1 A-over-A (score 1/2), 2 C-over-C (score 4/3), 1 G-over-G (score 1/2) and 3 T-over-T (score 9/4). Total score is –5/12.

But given two sequences, can we find an alignment that maximizes this score?

More General Substitution Scores?

How about the following *substitution-score matrix*?

	A	C	G	T
A	91	–114	–31	–123
C	–114	100	–125	–31
G	–31	–125	100	–114
T	–123	–31	–114	91

Optimal alignments under an arbitrary substitution-score matrix can be computed at essentially no penalty in computational time.

More General Substitution Scores?

How about a *position-specific scoring matrix*, that depends on the first sequence being aligned? For ACCTGAT we might want:

	A	C	C	T	G	A	T
A	91	–114	–63	–123	–31	33	27
C	–114	100	55	–31	–125	–42	–29
G	–31	–125	–81	–114	100	–8	–29
T	–123	–31	–112	91	–114	–49	27

Optimal alignments under these scores can be computed at essentially no penalty in computational time.

More General Gap Penalties?

Which alignment is preferable? (They have the same set of columns.)

```
ACAAT
A-A-T
```

or

```
ACAAT
A--AT
```

Gap Penalties (continued)

Let's subtract, say, 1 for each *gap*, i.e., run of consecutive dashes. Thus,

```
ACAAT
A-A-T
```

scores 1 less than does:

```
ACAAT
A--AT
```

Using such a *gap open penalty* roughly doubles the time for computing a highest-scoring alignment.

More General Alignment Scores?

Which alignment is preferable? (They have the same set of columns.)

```
ACTTCTCGAGGA...
||||||
ACTTCTTTTTTT...
```

or

```
ACCGTATGCGTA...
| | | | | |
ATCTTTTTCTTT...
```

What scoring rule makes the right distinction?

Let’s add 1 for each match that immediately follows another match. Thus,

```
ACTTCTCGAGGA...
||||||
ACTTCTTTTTTT...
```

scores 5 more than does:

```
ACCGTATGCGTA...
| | | | | |
ATCTTTTTCTTT...
```

Optimal alignments under these scores can be computed at only a small (say 10%) penalty in computational time.

More General Alignment Scores?

Which alignment is preferable? (Both have 12 matches.)

```
ACACACACACAC
ACACACACACAC
```

or

```
ACCGTATGCGTA
ACCGTATGCGTA
```

What scoring rule makes the right distinction?