Non-linear gap penalties

Waterman-Smith-Beyer (1976)
Gotoh (1982)

Rolf Backofen

Lehrstuhl für Bioinformatik
Institut für Informatik
Albert-Ludwigs-Universität Freiburg

Course Bioinformatics I

built on November 24, 2020

Gaps

- **Observation:** gaps are different in nature - given a fixed number of gaps, a "small number of long gaps" is biologically likelier than a "big number of small gaps"

Gaps

- **Observation:** gaps are different in nature - given a fixed number of gaps, a "small number of long gaps" is biologically likelier than a "big number of small gaps"

- **Problem:** with linear gap score not distinguishable:

$$\text{score}\begin{pmatrix}\texttt{AAA---}\\ \texttt{---TTT}\end{pmatrix} = \text{score}\begin{pmatrix}\texttt{A-A-A-}\\ \texttt{-T-T-T}\end{pmatrix}$$

Gaps

- **Observation:** gaps are different in nature - given a fixed number of gaps, a "small number of long gaps" is biologically likelier than a "big number of small gaps"

- **Problem:** with linear gap score not distinguishable:

$$\text{score}\begin{pmatrix}\texttt{AAA---}\\ \texttt{---TTT}\end{pmatrix} = \text{score}\begin{pmatrix}\texttt{A-A-A-}\\ \texttt{-T-T-T}\end{pmatrix}$$

- **Solution:** use explicit gap function $g : \mathbb{N}^+ \to \mathbb{R}$

Gaps

- **Observation:** gaps are different in nature - given a fixed number of gaps, a "small number of long gaps" is biologically likelier than a "big number of small gaps"
- **Problem:** with linear gap score not distinguishable:

$$\text{score}\begin{pmatrix}\texttt{AAA---}\\\texttt{---TTT}\end{pmatrix} = \text{score}\begin{pmatrix}\texttt{A-A-A-}\\\texttt{-T-T-T}\end{pmatrix}$$

- **Solution:** use explicit gap function $g : \mathbb{N}^+ \to \mathbb{R}$
- **Example:**

$$\begin{array}{ccccccccc} \overbrace{\texttt{A}\ \ \texttt{A}}^{\text{length 2}} & \texttt{A} & \overbrace{\texttt{-}\ \texttt{-}\ \texttt{-}}^{\text{length 3}} & \texttt{G} & \texttt{T} & \texttt{A} \\ \texttt{-}\ \ \texttt{-} & \texttt{A} & \texttt{T}\ \texttt{T}\ \texttt{T} & \texttt{G} & \texttt{T} & \underset{\text{length 1}}{\texttt{-}} \end{array}$$

$\Rightarrow$ Wanted: gap cost $g(2) + g(3) + g(1) \leq (2+3+1)*g(1)$

Gaps

- **Observation:** gaps are different in nature - given a fixed number of gaps, a "small number of long gaps" is biologically likelier than a "big number of small gaps"
- **Problem:** with linear gap score not distinguishable:

$$\text{score}\begin{pmatrix}\texttt{AAA---}\\ \texttt{---TTT}\end{pmatrix} = \text{score}\begin{pmatrix}\texttt{A-A-A-}\\ \texttt{-T-T-T}\end{pmatrix}$$

- **Solution:** use explicit gap function $g : \mathbb{N}^{+} \to \mathbb{R}$
- **Example:**

$$\begin{array}{l} \overbrace{\texttt{A}\ \texttt{A}}^{\text{length } 2}\ \texttt{A}\ \overbrace{\texttt{-}\ \texttt{-}\ \texttt{-}}^{\text{length } 3}\ \texttt{G}\ \texttt{T}\ \texttt{A} \\ \texttt{-}\ \texttt{-}\ \texttt{A}\ \texttt{T}\ \texttt{T}\ \texttt{T}\ \texttt{G}\ \texttt{T}\ \texttt{-} \end{array}$$

length 1

gap score difference translates to alignment scores $\Rightarrow$ distinguishable

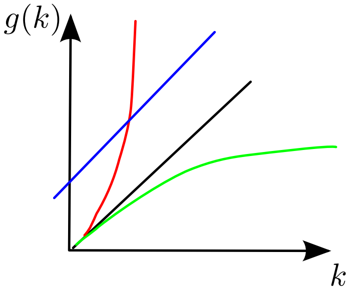
$\Rightarrow$ Wanted: gap cost $g(2) + g(3) + g(1) \leq (2+3+1)*g(1)$

Gap penalties

Definition (Gap penalty)

A *gap penalty* is a function $g : \mathbb{N}^+ \rightarrow \mathbb{R}$ that is subadditive, i.e.,

$$\forall k, l : g(k+l) \leq g(k) + g(l).$$

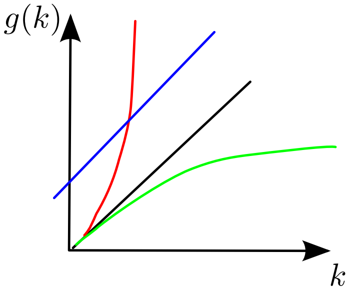


Definition (Gap penalty)

A *gap penalty* is a function $g : \mathbb{N}^+ \rightarrow \mathbb{R}$ that is subadditive, i.e.,

$$\forall k, l : g(k+l) \leq g(k) + g(l).$$

$g(k) = \alpha + \beta k^2$ ↯ not subadditive!
$\Rightarrow$unintended behaviour



Gap penalties

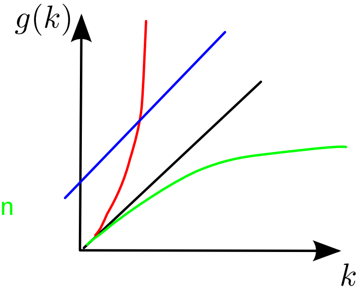
Definition (Gap penalty)

A *gap penalty* is a function $g : \mathbb{N}^+ \to \mathbb{R}$ that is subadditive, i.e.,

$$\forall k, l : g(k+l) \leq g(k) + g(l).$$

$g(k) = \alpha + \beta k^2$ ⚡ not subadditive!
$\Rightarrow$unintended behaviour

$g(k) = \alpha + \beta \ln(k)$ $\surd$
$\Rightarrow$ biologically the best approximation



UNI FREIBURG

Definition (Gap penalty)

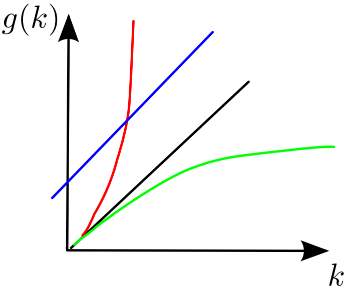
A *gap penalty* is a function $g : \mathbb{N}^{+} \rightarrow \mathbb{R}$ that is subadditive, i.e.,

$$\forall k, l : g(k + l) \leq g(k) + g(l).$$

$g(k) = \alpha + \beta k^2$ ⚡ not subadditive!
⇒unintended behaviour

$g(k) = \alpha + \beta \ln(k)$ √
⇒ biologically the best approximation

$g(k) = \alpha + \beta k$
⇒ affine, very common

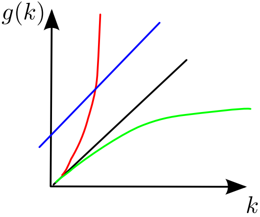


Idea: make initiation of gaps more expensive than extension of an existing gap

Definition (Affine gap penalty)

A gap penalty is called *affine* if there are $\alpha, \beta \in \mathbb{R}$ such that

$$g(k) = \alpha + \beta k$$



Gap penalty and alignment

- **Now:** How to calculate optimal alignments using a gap function?
- **Problem:** split like Needleman/Wunsch not working

$$w\begin{pmatrix} \mathtt{A} & \mathtt{A} & - & - \\ - & \mathtt{A} & \mathtt{G} & \mathtt{T} \end{pmatrix}$$

$$g(1) + w(\mathtt{A}, \mathtt{A}) + g(2)$$

Gap penalty and alignment

- **Now:** How to calculate optimal alignments using a gap function?
- **Problem:** split like Needleman/Wunsch not working

$$w\left(\begin{array}{ccc|c} \mathtt{A} & \mathtt{A} & \mathtt{-} & \mathtt{-} \\ \mathtt{-} & \mathtt{A} & \mathtt{G} & \mathtt{T} \end{array}\right) \qquad w\begin{pmatrix} \mathtt{A} & \mathtt{A} & \mathtt{-} \\ \mathtt{-} & \mathtt{A} & \mathtt{G} \end{pmatrix} + w\begin{pmatrix} \mathtt{-} \\ \mathtt{T} \end{pmatrix}$$

$$g(1) + w(\mathtt{A}, \mathtt{A}) + g(2) \neq \big(g(1) + w(\mathtt{A}, \mathtt{A}) + g(1)\big) + g(1)$$

Solution: more distinctions

1. Substitution: $\surd$

$$\begin{array}{lcr|l} a^\diamond & = & \ldots & a_i \\ b^\diamond & = & \ldots & b_j \end{array} \quad \Rightarrow \quad D_{i,j} = D_{i-1,j-1} + w(a_i, b_j)$$

Solution: more distinctions

1. Substitution: $\surd$

$$\begin{array}{lcl|l} a^\diamond & = & \dots & a_i \\ b^\diamond & = & \dots & b_j \end{array} \quad \Rightarrow \quad D_{i,j} = D_{i-1,j-1} + w(a_i, b_j)$$

2. Delete 1 position:

$$\begin{array}{lcll|l} a^\diamond & = & \dots & ? & a_i \\ b^\diamond & = & \dots & b_j & - \end{array} \quad \Rightarrow \quad D_{i,j} = D_{\underline{i-1},j} + g(1)$$

Solution: more distinctions

1. Substitution: $\surd$

$$\begin{array}{lcl|l} a^\diamond & = & \ldots & a_i \\ b^\diamond & = & \ldots & b_j \end{array} \quad \Rightarrow \quad D_{i,j} = D_{i-1,j-1} + w(a_i, b_j)$$

2. Delete 1 position:

$$\begin{array}{lcll|l} a^\diamond & = & \ldots & ? & a_i \\ b^\diamond & = & \ldots & b_j & - \end{array} \quad \Rightarrow \quad D_{i,j} = D_{\underline{i-1},j} + g(1)$$

3. Delete 2 positions:

$$\begin{array}{lcll|ll} a^\diamond & = & \ldots & ? & a_{i-1} & a_i \\ b^\diamond & = & \ldots & b_j & - & - \end{array} \quad \Rightarrow \quad D_{i,j} = D_{\underline{i-2},j} + g(2)$$

4. ...

$\Rightarrow$ Algorithm of Waterman-Smith-Beyer (1976)

Waterman, Smith, Beyer (1976)

Theorem (Waterman, Smith, and Beyer (1976))

Let $g : \mathbb{N} \to \mathbb{R}$ be a gap penalty and w be a distance function on $\Sigma \times \Sigma$. Let $a = a_1 \ldots a_n$ and $b = b_1 \ldots b_m$ be two words in Σ^*. We define $(D_{i,j})$ with $1 \leq i \leq n$ and $1 \leq j \leq m$ by

$$
\begin{aligned}
D_{0,0} &= 0, \\
D_{0,j} &= g(j), \\
D_{i,0} &= g(i), \\
D_{i,j} &= \min \left\{ \begin{array}{l} \min\limits_{1 \leq k \leq j} \{D_{i,j-k} + g(k)\}, \\ D_{i-1,j-1} + w(a_i, b_j), \\ \min\limits_{1 \leq k \leq i} \{D_{i-k,j} + g(k)\} \end{array} \right\}.
\end{aligned}
$$

Then $D_{i,j}$ = opt. prefix alignment score for $a_1 \ldots a_i$ and $b_1 \ldots b_j$.

Complexity of W-S-B's approach

$\Rightarrow$ on average a cell cost $O(n)$ for filling

$\Rightarrow$ total: $O(n^2)$ space and $O(n^3)$ time

- Example:
  - 2 RNA sequences with $n = 30\,000 = 3 \cdot 10^4$
  - **assume:** computer with 1 Ghz

$$
\begin{aligned}
&+ \text{ 1 operation per unit} \\
&\Rightarrow \quad \frac{27 \cdot 10^{12}\text{ops}}{10^9\text{ops}/s} \quad = \quad 27 \cdot 10^3 \text{ s} \\
&\qquad\qquad\qquad\qquad\quad / \; 3,600 \tfrac{s}{h} \\
&\qquad\qquad\qquad\quad \approx \quad 10.5 \text{ h}
\end{aligned}
$$

- How much time a *quadratic* algorithm would have taken?

Gotoh's idea

- W-S-B problem: arbitrarily long gaps tested in each step
- Solution: restrict to affine gap penalties $g(k) = \alpha + \beta k$

Gotoh's idea

- W-S-B problem: arbitrarily long gaps tested in each step
- Solution: restrict to affine gap penalties $g(k) = \alpha + \beta k$

Analyzing gap case:

$$\begin{array}{lcc|l} a^\diamond & = & \ldots & a_i \\ b^\diamond & = & \ldots & - \end{array} \quad \Rightarrow \quad \textbf{look at subcases}$$

Gotoh's idea

- W-S-B problem: arbitrarily long gaps tested in each step
- Solution: restrict to affine gap penalties $\quad g(k) = \alpha + \beta k$

Analyzing gap case:

$$\begin{array}{lcrl} a^\diamond & = & \dots & \mid a_i \\ b^\diamond & = & \dots & \mid - \end{array} \quad \Rightarrow \quad \textbf{look at subcases}$$

1. $$\begin{array}{lr|l} \dots & ? & a_i \\ \dots & b_j & - \end{array} \quad \Rightarrow \quad D_{i,j} = D_{i-1,j} + g(1)$$

Gotoh's idea

- W-S-B problem: arbitrarily long gaps tested in each step
- Solution: restrict to affine gap penalties $g(k) = \alpha + \beta k$

Analyzing gap case:

$$\begin{array}{lcc|l} a^\diamond & = & \dots & a_i \\ b^\diamond & = & \dots & - \end{array} \Rightarrow \textbf{look at subcases}$$

1. $$\begin{array}{cc|c} \dots & ? & a_i \\ \dots & b_j & - \end{array} \Rightarrow D_{i,j} = D_{i-1,j} + g(1)$$

2. $$\underbrace{\begin{array}{cc|c} \dots & a_{i-1} & a_i \\ \dots & - & - \end{array}}_{k \text{ gaps}} \Rightarrow \begin{aligned} D_{i,j} &= \star - g(k-1) + g(k) \\ &= \star - \alpha - (k-1)\beta + \alpha + k\beta \\ &= \star + \beta \end{aligned}$$

$(\star$: best score ending in gap $\begin{smallmatrix} a_{i-1} \\ - \end{smallmatrix})$

...analogously for end gaps in $a^\diamond$

Gotoh's idea

$\Rightarrow$ recursion cases:

a. no gap $\Rightarrow +w(..)$
b. starting a new gap $\Rightarrow +g(1)$
c. elongate an existing gap $\Rightarrow +\beta$

$\Rightarrow$ length of the gap doesn't matter

$\Rightarrow$ saving time because:

- W–S–B: test with all possible gap lengths
- Gotoh: just add β if a gap is elongated

Gotoh's idea

- $\Rightarrow$ recursion cases:
  a. no gap $\Rightarrow +w(..)$
  b. starting a new gap $\Rightarrow +g(1)$
  c. elongate an existing gap $\Rightarrow +\beta$

- $\Rightarrow$ length of the gap doesn't matter

- $\Rightarrow$ saving time because:
  - W–S–B: test with all possible gap lengths
  - Gotoh: just add β if a gap is elongated

- **Comment:** if gap penalty is not affine (e.g. $g(k) = \alpha + \beta \cdot \ln(k)$) then

$$\begin{aligned} D_{i,j} &= \star - g(k-1) + g(k) \\ &= \dots \end{aligned}$$

Gotoh's idea

$\Rightarrow$ recursion cases:
a. no gap $\Rightarrow +w(..)$
b. starting a new gap $\Rightarrow +g(1)$
c. elongate an existing gap $\Rightarrow +\beta$

$\Rightarrow$ length of the gap doesn't matter

$\Rightarrow$ saving time because:

- W–S–B: test with all possible gap lengths
- Gotoh: just add β if a gap is elongated

- **Comment:** if gap penalty is not affine (e.g. $g(k) = \alpha + \beta \cdot \ln(k)$) then

$$\begin{aligned} D_{i,j} &= \star - g(k-1) + g(k) \\ &= \star - \alpha - \ln(k-1) + \alpha + \ln(k) \\ &= \star + \ln(k) - \ln(k-1) \\ &= \star + \ln(\tfrac{k}{k-1}) \end{aligned}$$

- $\Rightarrow$ depends on k $\Rightarrow$ Gotoh's idea doesn't work

Gotoh matrices

$\Rightarrow$ further matrices needed

- $(D_{i,j})$ cost for alignment of prefixes $(a_1 \ldots a_i, b_1 \ldots b_j)$
- $(P_{i,j})$ cost for alignment of prefixes $(a_1 \ldots a_i, b_1 \ldots b_j)$ that ends with a gap in $b^\diamond$ (i.e., last column is $\binom{a_i}{-}$))
- $(Q_{i,j})$ cost for alignment of prefixes $(a_1 \ldots a_i, b_1 \ldots b_j)$ that ends with a gap in $a^\diamond$ (i.e., last column is $\binom{-}{b_j}$))

Gotoh (1982)

- affine gap penalty $g(k) = \alpha + k\beta$; distance function $w : \Sigma \times \Sigma \to \mathbb{R}$
- recursive definition of matrices $(D_{i,j})$, $(P_{i,j})$, and $(Q_{i,j})$:

Gotoh (1982)

- affine gap penalty $g(k) = \alpha + k\beta$; distance function $w : \Sigma \times \Sigma \to \mathbb{R}$
- recursive definition of matrices $(D_{i,j})$, $(P_{i,j})$, and $(Q_{i,j})$:

$$D_{i,j} \quad = \quad \min \left\{ \begin{array}{l} D_{i-1,j-1} + w(a_i, b_j) \\ P_{i,j} \\ Q_{i,j} \end{array} \right\},$$

with $i, j \geq 1$, where for $1 \leq i \leq |a|$ and $1 \leq j \leq |b|$,

Gotoh (1982)

- affine gap penalty $g(k) = \alpha + k\beta$; distance function $w : \Sigma \times \Sigma \to \mathbb{R}$
- recursive definition of matrices $(D_{i,j})$, $(P_{i,j})$, and $(Q_{i,j})$:

$$D_{i,j} = \min \left\{ \begin{array}{l} D_{i-1,j-1} + w(a_i, b_j) \\ P_{i,j} \\ Q_{i,j} \end{array} \right\},$$

with $i, j \geq 1$, where for $1 \leq i \leq |a|$ and $1 \leq j \leq |b|$,

$$P_{i,j} = \min \left\{ \begin{array}{l} D_{i-1,j} + g(1) \\ P_{i-1,j} + \beta \end{array} \right\}$$

$$Q_{i,j} = \min \left\{ \begin{array}{l} D_{i,j-1} + g(1) \\ Q_{i,j-1} + \beta \end{array} \right\}$$

Gotoh (1982)

- affine gap penalty $g(k) = \alpha + k\beta$; distance function $w : \Sigma \times \Sigma \to \mathbb{R}$
- recursive definition of matrices $(D_{i,j})$, $(P_{i,j})$, and $(Q_{i,j})$:

$$D_{i,j} = \min \left\{ \begin{array}{l} D_{i-1,j-1} + w(a_i, b_j) \\ P_{i,j} \\ Q_{i,j} \end{array} \right\},$$

with $i, j \geq 1$, where for $1 \leq i \leq |a|$ and $1 \leq j \leq |b|$,

$$P_{i,j} = \min \left\{ \begin{array}{l} D_{i-1,j} + g(1) \\ P_{i-1,j} + \beta \end{array} \right\}$$

$$Q_{i,j} = \min \left\{ \begin{array}{l} D_{i,j-1} + g(1) \\ Q_{i,j-1} + \beta \end{array} \right\}$$

order of calculation: $P_{i,j}$, $Q_{i,j}$ before $D_{i,j}$

Gotoh (1982) - Initialization

- D as usual:
$$\begin{aligned} D_{0,0} &= 0, \\ D_{0,j} &= g(j), \\ D_{i,0} &= g(i) \end{aligned}$$

Gotoh (1982) - Initialization

- D as usual: $$\begin{aligned} D_{0,0} &= 0, \\ D_{0,j} &= g(j), \\ D_{i,0} &= g(i) \end{aligned}$$
- P:
  - recursion only on the first index $(P_{i,j} \rightarrow P_{i-1,j} \rightarrow \ldots \rightarrow P_{0,j})$
  - **hence:** only initialization for $P_{0,j}$ $\Rightarrow P_{i,0}$ not used

Gotoh (1982) - Initialization

- D as usual: $$\begin{aligned} D_{0,0} &= 0, \\ D_{0,j} &= g(j), \\ D_{i,0} &= g(i) \end{aligned}$$
- P:
  - recursion only on the first index $(P_{i,j} \to P_{i-1,j} \to \ldots \to P_{0,j})$
  - **hence:** only initialization for $P_{0,j}$ $\Rightarrow P_{i,0}$ not used
  - **but:** $P_{0,j}$ is best alignment of ϵ and $b_1 \ldots b_j$ that ends with gap in $b^\diamond$ $\Rightarrow$ the only possible alignment would be:

$$\begin{array}{cccccc} - & - & \ldots & - & - & - \\ b_1 & b_2 & \ldots & b_{j-1} & b_j & - \end{array}$$

$\Updownarrow$

disallowed in alignments!

$\Rightarrow P_{0,j} = \infty$

Gotoh (1982) - Initialization

- D as usual:
$$\begin{aligned} D_{0,0} &= 0, \\ D_{0,j} &= g(j), \\ D_{i,0} &= g(i) \end{aligned}$$

- P:
  - recursion only on the first index $(P_{i,j} \rightarrow P_{i-1,j} \rightarrow \ldots \rightarrow P_{0,j})$
  - **hence:** only initialization for $P_{0,j}$ $\Rightarrow$ $P_{i,0}$ not used
  - **but:** $P_{0,j}$ is best alignment of ϵ and $b_1 \ldots b_j$ that ends with gap in $b^\diamond$ $\Rightarrow$ the only possible alignment would be:

$$\begin{array}{cccccc} - & - & \ldots & - & - & - \\ b_1 & b_2 & \ldots & b_{j-1} & b_j & - \end{array}$$

$$\Updownarrow$$

disallowed in alignments!

$$\Rightarrow P_{0,j} = \infty$$

- Q analogously ...

Example

- given: $a = \texttt{CC}$ and $b = \texttt{ACCT}$.
- cost functions:
  - substitutions: $w(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{else} \end{cases}$
  - gap penalty: $g(k) = 4 + k$ $(\beta = 1)$.
- **wanted:** optimal alignment using Gotoh

Example - filling of matrices

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	–				
C	–				

$(Q_{i,j}) =$

		A	C	C	T
	0	–	–	–	–
C	∞				
C	∞				

$(D_{i,j}) =$

		A	C	C	T
	0	5	6	7	8
C	5				
C	6				

Example - filling of matrices

$$(P_{i,j}) = \begin{array}{c|c|c|c|c|c|}
 & & A & C & C & T \\ \hline
 & 0 & \infty & \infty & \infty & \infty \\ \hline
C & - & 10 & & & \\ \hline
C & - & & & & \\ \hline
\end{array}$$

$$P_{1,1} = \min \begin{cases} D_{1,0} + g(1) & (= 5 + 5 = 10) \\ P_{1,0} + \beta & (= \infty + 1 = \infty) \end{cases} = 10$$

$$(Q_{i,j}) = \begin{array}{c|c|c|c|c|c|}
 & & A & C & C & T \\ \hline
 & 0 & - & - & - & - \\ \hline
C & \infty & & & & \\ \hline
C & \infty & & & & \\ \hline
\end{array}$$

$$(D_{i,j}) = \begin{array}{c|c|c|c|c|c|}
 & & A & C & C & T \\ \hline
 & 0 & 5 & 6 & 7 & 8 \\ \hline
C & 5 & & & & \\ \hline
C & 6 & & & & \\
\end{array}$$

Example - filling of matrices

$$(P_{i,j}) = \begin{array}{c|c|c|c|c|c|} & & A & C & C & T \\ \hline & 0 & \infty & \infty & \infty & \infty \\ \hline C & - & & & & \\ \hline C & - & & & & \\ \hline \end{array}$$

$$(Q_{i,j}) = \begin{array}{c|c|c|c|c|c|} & & A & C & C & T \\ \hline & 0 & - & - & - & - \\ \hline C & \infty & 10 & & & \\ \hline C & \infty & & & & \\ \hline \end{array}$$

$$(D_{i,j}) = \begin{array}{c|c|c|c|c|c|} & & A & C & C & T \\ \hline & 0 & 5 & 6 & 7 & 8 \\ \hline C & 5 & & & & \\ \hline C & 6 & & & & \\ \end{array}$$

$$Q_{1,1} = \min \left\{ \begin{array}{ll} D_{0,1} + g(1) & (= 5 + 5 \ = 10) \\ Q_{0,1} + \beta & (= \infty + 1 = \infty) \end{array} \right\}$$
$$= 10$$

Example - filling of matrices

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	–	10			
C	–				

$(Q_{i,j}) =$

		A	C	C	T
	0	–	–	–	–
C	∞	10			
C	∞				

$(D_{i,j}) =$

		A	C	C	T
	0	5	6	7	8
C	5	1			
C	6				

$$D_{1,1} = \min \left\{ \begin{array}{ll} D_{0,0} + w(C,A) & (= 1) \\ P_{1,1} & (= 10) \\ Q_{1,1} & (= 10) \end{array} \right\}$$

$$= 1$$

Example - filling of matrices

$$(P_{i,j}) = \begin{array}{c|c|c|c|c|c}
 & & A & C & C & T \\ \hline
 & 0 & \infty & \infty & \infty & \infty \\ \hline
C & - & 10 & & & \\ \hline
C & - & & & &
\end{array}$$

$$P_{1,1} = \min \left\{ \begin{array}{ll} D_{1,0} + g(1) & (= 5 + 5 = 10) \\ P_{1,0} + \beta & (= \infty + 1 = \infty) \end{array} \right\}$$
$$= 10$$

$$(Q_{i,j}) = \begin{array}{c|c|c|c|c|c}
 & & A & C & C & T \\ \hline
 & 0 & - & - & - & - \\ \hline
C & \infty & 10 & & & \\ \hline
C & \infty & & & &
\end{array}$$

$$Q_{1,1} = \min \left\{ \begin{array}{ll} D_{0,1} + g(1) & (= 5 + 5 = 10) \\ Q_{0,1} + \beta & (= \infty + 1 = \infty) \end{array} \right\}$$
$$= 10$$

$$(D_{i,j}) = \begin{array}{c|c|c|c|c|c}
 & & A & C & C & T \\ \hline
 & 0 & 5 & 6 & 7 & 8 \\ \hline
C & 5 & 1 & & & \\ \hline
C & 6 & & & &
\end{array}$$

$$D_{1,1} = \min \left\{ \begin{array}{ll} D_{0,0} + w(C, A) & (= 1) \\ P_{1,1} & (= 10) \\ Q_{1,1} & (= 10) \end{array} \right\}$$
$$= 1$$

Example - traceback

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	−	10	11	12	13
C	−	6	10	11	13

Final matrices

$(Q_{i,j}) =$

		A	C	C	T
	0	−	−	−	−
C	∞	10	6	7	8
C	∞	11	11	6	7

$(D_{i,j}) =$

		A	C	C	T
	0	5	6	7	8
C	5	1	5	6	8
C	6	6	1	5	7

Example - traceback

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	−	10	11	12	13
C	−	6	10	11	13

$(Q_{i,j}) =$

		A	C	C	T
	0	−	−	−	−
C	∞	10	6	7	8
C	∞	11	11	6	7

$(D_{i,j}) =$

		A	C	C	T
	0	5	6	7	8
C	5	1	5	6	8
C	6	6	1	5	7

Final matrices and tracebacks

1. $D \nwarrow \quad D \nwarrow \quad D \leftarrow \quad Q \leftarrow \quad Q \bullet$

```
C   C   -   -
A   C   C   T
```

Example - traceback

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	–	10	11	12	13
C	–	6	10	11	13

$(Q_{i,j}) =$

		A	C	C	T
	0	–	–	–	–
C	∞	10	6	7	8
C	∞	11	11	6	7

$(D_{i,j}) =$

		A	C	C	T
	0	5	6	7	8
C	5	1	5	6	8
C	6	6	1	5	7

Final matrices and tracebacks

1. $D\nwarrow$ $D\nwarrow$ $D\leftarrow$ $Q\leftarrow$ $Q\bullet$

```
C C - -
A C C T
```

2. $D\leftarrow$ $D\leftarrow$ $D\nwarrow$ $D\nwarrow$

```
- - C C
A C C T
```

Example - traceback

$(P_{i,j}) =$

		A	C	C	T
	0	∞	∞	∞	∞
C	**–**	**10**	**11**	**12**	**13**
C	**–**	**6**	**10**	**11**	**13**

$(Q_{i,j}) =$

		A	C	C	T
	0	**–**	**–**	**–**	**–**
C	∞	**10**	**6**	**7**	**8**
C	∞	**11**	**11**	D **6**	**7** Q

$(D_{i,j}) =$

		A	C	C	T
	0	D **5**	D **6**	**7**	**8**
C	**5**	D **1**	**5**	D **6**	**8**
C	**6**	**6**	D **1**	**5**	D **7** Q

Final matrices and tracebacks

1. $D\nwarrow$ $D\nwarrow$ $D\leftarrow$ $Q\leftarrow$ $Q\bullet$

```
C  C  -  -
A  C  C  T
```

2. $D\leftarrow$ $D\leftarrow$ $D\nwarrow$ $D\nwarrow$

```
-  -  C  C
A  C  C  T
```

Summary

- Problem: long gaps are often single evol. events

  $\Rightarrow$ linear gap scoring leads to gap distribution

  $\Rightarrow$ too high penalty for long gaps (need subadditive gap scoring)

Summary

- Problem: long gaps are often single evol. events
  $\Rightarrow$ linear gap scoring leads to gap distribution
  $\Rightarrow$ too high penalty for long gaps (need subadditive gap scoring)

- Waterman-Smith-Beyer: optimization with subadditive gap score
  $\Rightarrow$ explicit consideration of every gap length
  $\Rightarrow$ increases runtime complexity to $O(n^3)$ compared to linear gap cost

Summary

- Problem: long gaps are often single evol. events
  $\Rightarrow$ linear gap scoring leads to gap distribution
  $\Rightarrow$ too high penalty for long gaps (need subadditive gap scoring)

- Waterman-Smith-Beyer: optimization with subadditive gap score
  $\Rightarrow$ explicit consideration of every gap length
  $\Rightarrow$ increases runtime complexity to $O(n^3)$ compared to linear gap cost

- Gotoh: optimization with affine gap score
  $\Rightarrow$ distinction of gap start and extension
  $\Rightarrow$ auxiliary matrices needed for gap handling
  $\Rightarrow$ runtime complexity again $O(n^2)$