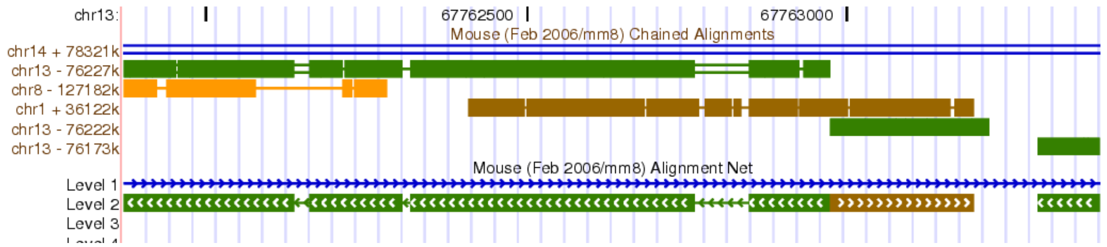
Sequence Alignment (zarovnávanie sekvencií) 1/2

Tomáš Vinař

October 7, 2021



[Durbin et al., 1998, chapter 2]

Problem: Local alignment

```
ggcccttggagttgactgtcctgctgctccttgagg
ccattctcagagagaggaagtggcctcattttaatc
cgcttcccacagccttgtcctttccagacccatggg
agagggaggggctgagggtgtggctgagcccaccca
agtcacgcgtcactctgcaggtccctctcccccaag
gccgtggccttgggagcccgtggatcccagtgagtg
acgcctccacccccgccctactcgggcagtttaac
ccttgttgttcacttgcagacatcgtgaacacggcc
cggcccgacgagaaggccataatgacctatgtgtcc
agcttctaccatgccttttcaggagcgcagaaggta
ccgagcagggccaggcaggccctcctcgccgccacc
gcgcaatgccgccgctgcctctcgcctcccgtgctc
acctcatttctcttgcagacggcagtggcctctctc
caactggaagccacccccagctccct...
```

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Input: two sequences

Problem: Local alignment

```
ggcccttggagttgactgtcctgctgctccttgagg
ccattctcagagagaggaagtggcctcattttaatc
cgcttcccacagccttgtcctttccagacccatggg
agagggaggggctgagggtgtggctgagcccaccca
agtcacgcgtcactctgcaggtccctctcccccaag
gccgtggccttgggagcccgtggatcccagtgagtg
acgcctccacccccgccctactcgggcagtttaac
ccttgttgttcacttgcagacatcgtgaacacggcc
cggcccgacgagaaggccataatgacctatgtgtcc
agcttctaccatgcctttttcaggagcgcagaaggta
ccgagcagggccaggcaggccctcctcgccgccacc
gcgcaatgccgccgctgcctctcgcctcccgtgctc
acctcatttctcttgcagacggcagtggcctctctc
caactggaagccacccccagctccct...
```

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Output: similar regions (in the form of an alignment)

```
CCCGACGAGAAGGCCATAATGACCTATGTGTCCAGCTTCTACCATGCCTTT
|| ||||||||| ||||    ||||| |||  || || ||| ||   ||||
CCGGACGAGAAGTCCAT---CACCTACGTGGTCACCTACTATCACTACTTT
```

Insert dashes (gaps) so that corresponding bases in the same column.

A good alignment has many aligned matching bases, few gaps.

What are alignments good for?

- **Orientation in large sequence databases.**
  Genbank has more 3 TB of whole genome sequences.
  E.g.: from which genome (and which part) comes a given sequence?
- **Determine function (e.g. of a protein).**
  Similar sequences often have the same or similar function.
- **Evolutionary studies.**
  Search for homologs, sequences which have evolved from the same common ancestor.
  In the ideal case, gaps correspond to insertions and deletions, aligned bases to conserved bases and substitutions.
- **Finding genes and other functional elements.**
  These often change slower than other sequences.

Sequence alignment as an optimization problem

Goal of the sequence alignment: find pairs of homologous bases (coming from a common ancestor)

Modeling phase: choose a scoring scheme such that
– real alignments have high score
– false positives have low score

Optimization phase:
given two input sequences find the highest scoring alignment
– focus on computational efficiency

Problem formulation

Set up a **scoring scheme** for alignments

e.g. match +1, mismatch -1, gap -1

```
GAGAAGGCCATAATGACCTATGTGTCCAGCT
|||||| ||||    ||||| |||  || ||
GAGAAGTCCAT---CACCTACGTGGTCACCT
```

22 matches, 6 mismatches, 3 gaps $\rightarrow$ score 13.

In practice we often use more complex scoring schemes.

Problem 1: global alignment

Input: sequences $X = x_1 x_2 \dots x_n$ and $Y = y_1 y_2 \dots y_m$.

Output: alignment of X and Y with the highest score

Problem 2: local alignment

Input: sequences $X = x_1 x_2 \dots x_n$ and $Y = y_1 y_2 \dots y_m$.

Output: alignment of substrings $x_i \dots x_j$ and $y_k \dots y_\ell$ with highest score

Dynamic programming for global alignment (Needleman, Wunsch 1970)

Subproblem $A[i,j]$: highest score of a global alignment of $x_1 x_2 \ldots x_i$ a $y_1 y_2 \ldots y_j$

One of the strings has length 0: the other string is aligned to gaps
$A[0,j] = -j$, $A[i,0] = -i$

General case $i > 0$, $j > 0$:
if $x_i = y_j$ are aligned $A[i,j] = A[i-1,j-1] + 1$
if $x_i \neq y_j$ are aligned $A[i,j] = A[i-1,j-1] - 1$
if x_i is aligned to a gap $A[i,j] = A[i-1,j] - 1$
if y_j is aligned to a gap $A[i,j] = A[i,j-1] - 1$

$$\begin{array}{cc} x_1 \ldots x_{i-1} & x_i \\ \underbrace{y_1 \ldots y_{j-1}}_{A[i-1,j-1]} & \underbrace{y_j}_{\pm 1} \end{array} \qquad \begin{array}{cc} x_1 \ldots x_{i-1} & x_i \\ \underbrace{y_1 \ldots y_j}_{A[i-1,j]} & \underbrace{-}_{-1} \end{array} \qquad \begin{array}{cc} x_1 \ldots x_i & - \\ \underbrace{y_1 \ldots y_{j-1}}_{A[i,j-1]} & \underbrace{y_j}_{-1} \end{array}$$

Dynamic programming for global alignment

Subproblem $A[i,j]$: highest score of a global alignment of $x_1 x_2 \ldots x_i$ a $y_1 y_2 \ldots y_j$

General case $i > 0$, $j > 0$:

if $x_i = y_j$ are aligned $A[i,j] = A[i-1,j-1] + 1$
if $x_i \neq y_j$ are aligned $A[i,j] = A[i-1,j-1] - 1$
if x_i is aligned to a gap $A[i,j] = A[i-1,j] - 1$
if y_j is aligned to a gap $A[i,j] = A[i,j-1] - 1$

Recurrence:

$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i, y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

where $s(x,y) = 1$ if $x = y$ and $s(x,y) = -1$ if $x \neq y$

Global alignment example

CATGTCGTA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	?						
G	-4										
T	-5										
C	-6										
G	-7										
T	-8										
A	-9										

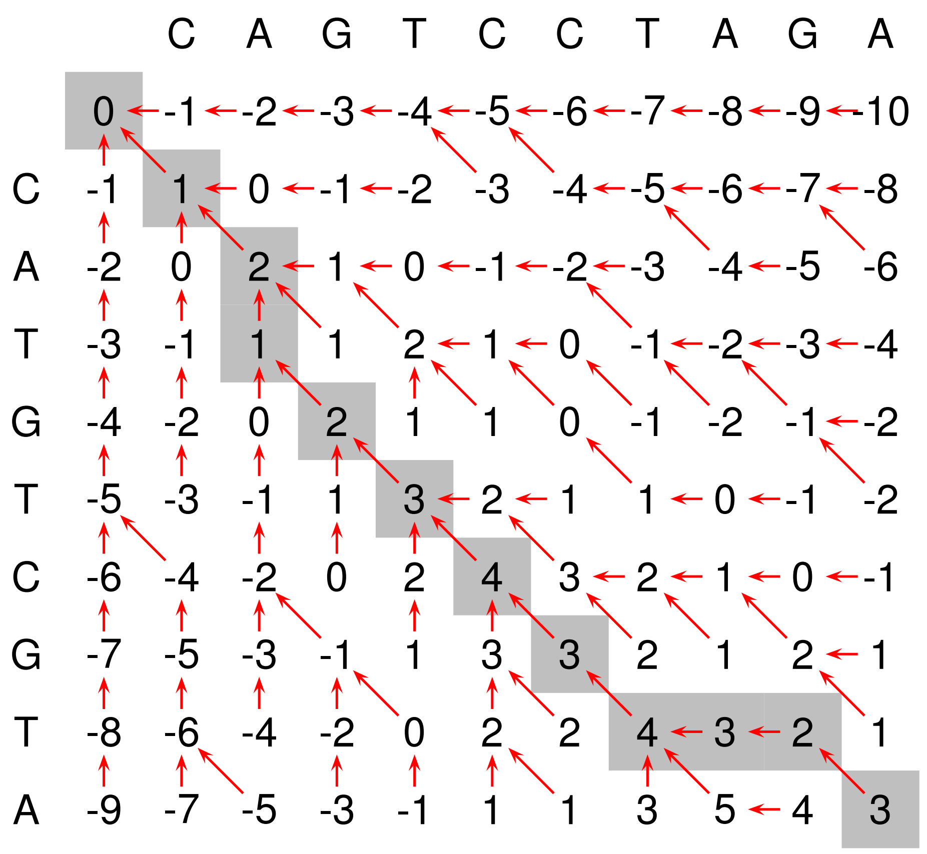
$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i,y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Global alignment example

CATGTCGTA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	2	1	0	-1	-2	-3	-4
G	-4	-2	0	2	1	1	0	-1	-2	-1	-2
T	-5	-3	-1	1	3	2	1	1	0	-1	-2
C	-6	-4	-2	0	2	4	3	2	1	0	-1
G	-7	-5	-3	-1	1	3	3	2	1	2	1
T	-8	-6	-4	-2	0	2	2	4	3	2	1
A	-9	-7	-5	-3	-1	1	1	3	5	4	3

How to get the alignment?



		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	2	1	0	-1	-2	-3	-4
G	-4	-2	0	2	1	1	0	-1	-2	-1	-2
T	-5	-3	-1	1	3	2	1	1	0	-1	-2
C	-6	-4	-2	0	2	4	3	2	1	0	-1
G	-7	-5	-3	-1	1	3	3	2	1	2	1
T	-8	-6	-4	-2	0	2	2	4	3	2	1
A	-9	-7	-5	-3	-1	1	1	3	5	4	3

```
CA-GTCCTAGA
CATGTCGT--A
```

Dynamic programming for local alignment (Smith, Waterman 1981)

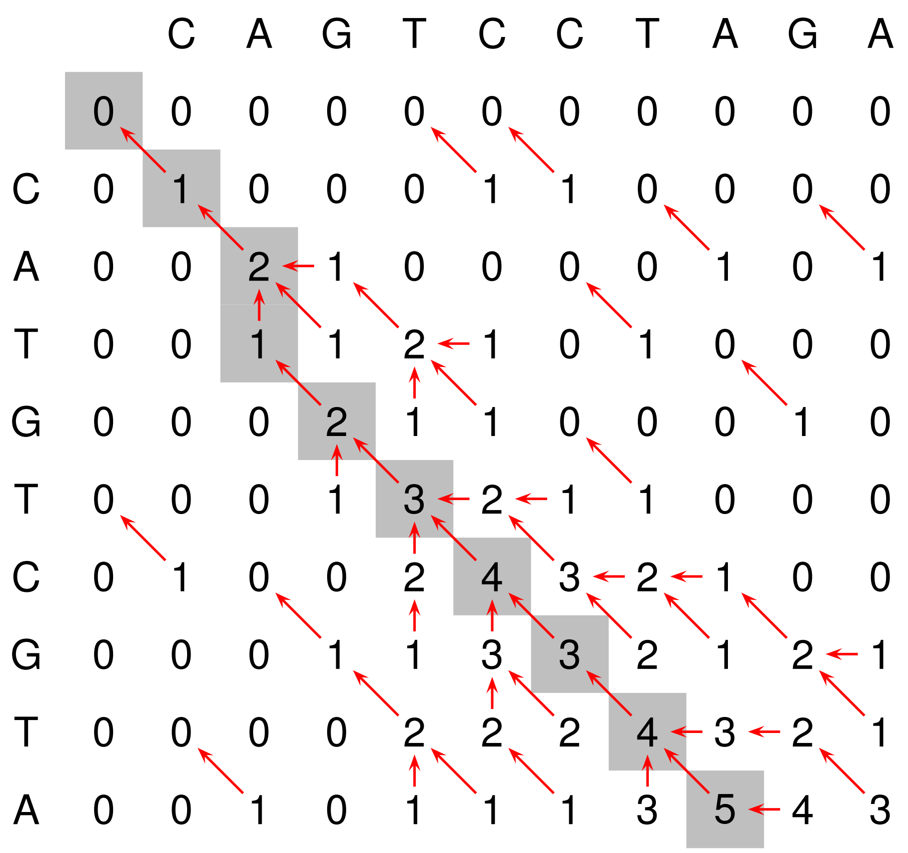
Subproblem $A[i,j]$: highest score of a local alignment of $x_1x_2\ldots x_i$ a $y_1y_2\ldots y_j$ that contains both x_i and y_j or is empty

One of the strings has length 0: $A[0,j] = A[i,0] = 0$ (empty aln.)

General case $i > 0$, $j > 0$:
if x_i and y_j are aligned $A[i,j] = A[i-1,j-1] + s(x_i,y_j)$
if x_i is aligned to a gap $A[i,j] = A[i-1,j] - 1$
if y_j is aligned to a gap $A[i,j] = A[i,j-1] - 1$
if x_i and y_j are not part of alignment with a positive score $A[i,j] = 0$

Recurrence: $$A[i,j] = \max \begin{cases} 0, \\ A[i-1,j-1] + s(x_i,y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Example of local alignment



		C	A	G	T	C	C	T	A	G	A
	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	1	1	0	0	0	0
A	0	0	2	1	0	0	0	0	1	0	1
T	0	0	1	1	2	1	0	1	0	0	0
G	0	0	0	2	1	1	0	0	0	1	0
T	0	0	0	1	3	2	1	1	0	0	0
C	0	1	0	0	2	4	3	2	1	0	0
G	0	0	0	1	1	3	3	2	1	2	1
T	0	0	0	0	2	2	2	4	3	2	1
A	0	0	1	0	1	1	1	3	5	4	3

```
CA-GTCCTA
CATGTCGTA
```

More complex scoring schemes

Problems of the $+1, -1$ scoring scheme:

- Is really one mismatch or gap that bad compared to a single match?
- How to score protein alignments?
  (20 element alphabet $\approx$ 200 parameters)

Goal of the scoring scheme:

- We want to distinguish better alignments from worse:
  – Which arrangements of gaps are more meaningful?
- We want to know if an alignment has a biological meaning:
  – Are the two sequences homologs or unrelated?

Probabilistic scoring scheme (the first attempt)

Assume X and Y are **correctly aligned homologs**

$a =$ probability that two bases form a **match**

$b =$ probability that two bases form a **mismatch**

$c =$ probability that a base is aligned to a **gap**

$a + b + c = 1$

Probability of alignment A:

```
GAGAAGGCCATAATGACCTATGTGTCCAGCT
|||||| ||||    ||||| |||  || ||
GAGAAGTCCAT---CACCTACGTGGTCACCT
```

$$\Pr(A) = a^{22} b^{6} c^{3}$$

Which alignment is more likely?

```
CACA
|  |
CCAA
```

$$\Pr(A) = a^2 b^2$$

```
CACA-
| ||
C-CAA
```

$$\Pr(A) = a^3 c^2$$

Probabilistic scoring scheme (the first attempt)

Take logarithm to change multiplication into addition
we can use S.-W. or N.-W. dynamic programming algorithms

$\Pr(A) = a^{22} b^6 c^3$
$\log \Pr(A) = 22 \log a + 6 \log b + 3 \log c$

Score: Match: $\log a$ Mismatch: $\log b$ Gap: $\log c$

Disadvantage of this scheme:

- Score always negative $\Rightarrow$ how to do local alignment?
- Hard to compare different pairs of sequences

Scoring scheme based on two probabilistic models

Compare models H and R: “log likelihood ratio”

$$\log \frac{\Pr(X, Y \mid H)}{\Pr(X, Y \mid R)}$$

- Two sequences are **homologs**

  $\Rightarrow$ likelihood ratio much higher than 1

  $\Rightarrow$ positive score

- Two **unrelated** sequences

  $\Rightarrow$ likelihood ratio much lower than 1

  $\Rightarrow$ negative score

Scoring scheme based on two probabilistic models

(Ignore gaps for now)

Model H: Sequences X and Y are **correctly aligned homologs**

$\Pr(X, Y \mid H) = \prod_{i=1}^{n} p(x_i, y_i)$

$p(x_i, y_i)$: probability that alignment contains aligned bases x_i and y_i

Model R: Sequences X and Y are unrelated

$\Pr(X, Y \mid R) = \left(\prod_{i=1}^{n} p(x_i)\right) \left(\prod_{i=1}^{n} p(y_i)\right)$

$p(x_i)$: probability of occurrence of x_i in a sequence

Compare models H and R: "log likelihood ratio"

$$\log \frac{\Pr(X, Y \mid H)}{\Pr(X, Y \mid R)}$$

Scoring scheme based on two probabilistic models

$\Pr(X, Y \mid H) = \prod_{i=1}^{n} p(x_i, y_i)$

$\Pr(X, Y \mid R) = \left(\prod_{i=1}^{n} p(x_i)\right) \left(\prod_{i=1}^{n} p(y_i)\right)$

$$\log \frac{\Pr(X, Y \mid H)}{\Pr(X, Y \mid R)} = \log \frac{\prod_{i=1}^{n} p(x_i, y_i)}{\left(\prod_{i=1}^{n} p(x_i)\right) \left(\prod_{i=1}^{n} p(y_i)\right)} = \sum_{i=1}^{n} \log \frac{p(x_i, y_i)}{p(x_i)p(y_i)}$$

score for aligning bases x and y:

$$s(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

BLOSUM62 protein scoring matrix

BLOcks of aminoacid **SU**bstitution **M**atrix; Henikoff, Henikoff 1992

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

- Choose **biologically relevant protein alignments** (BLOCKS)
- Only pairs with identity at most 62%
- $p(x,y)$: how often we see amino acids x and y aligned
- $p(x)$: how often we see amino acid x

- **Score for a pair of amino acids x and y**: $\log \dfrac{p(x,y)}{p(x)p(y)}$
- multiply by a constant and round to integers:
  - to avoid too big rounding error
  - integers allow faster computation

More complex scoring: Affine gap scores

```
CCCGACGAGAAGGCCATAATGACCTATGTGTCCAGCTTCTACCATGCCTTT
|| ||||||||| ||||    ||||| |||  || || ||| ||   ||||
CCGGACGAGAAGTCCAT---CACCTACGTGGTCACCTACTATCACTACTTT
```

Several consecutive gaps likely originated in a single mutation rather than each independently.

Penalty for starting a gap (gap opening cost) o,
Penalty for each next gap symbol (gap extension cost) e.
Gap of length g has penalty $o + e(g - 1)$.
We choose $o < e$ (i.e. $|o| > |e|$).

Default settings of blastn: match +2, mismatch -3, $o = -5$, $e = -2$.
Example above: 22 matches, 6 mismatches, 1 gap of length 3
$\rightarrow$ score $2 \cdot 22 - 3 \cdot 6 - 5 - 2 \cdot 2 = 16$.

Summary

- Global and local alignments
- Needleman-Wunsch and Smith-Waterman algorithms
- Scoring schemes for alignments based on comparing likelihoods
- Protein BLOSUM scoring matrix
- Affine gap penalties

Problems to think about:

1. **Running time of Smith-Waterman:** $O(nm)$
   n - length of the first sequence
   m - length of the second sequence
   Local alignments between human and mouse?

2. We found an alignment with score 14
   Is this a good score or is it a score that would appear just by chance?