

Efektywne algorytmy do porównań sekwencji

Bartek Wilczyński

13. marca, 2018

How sequences evolve?

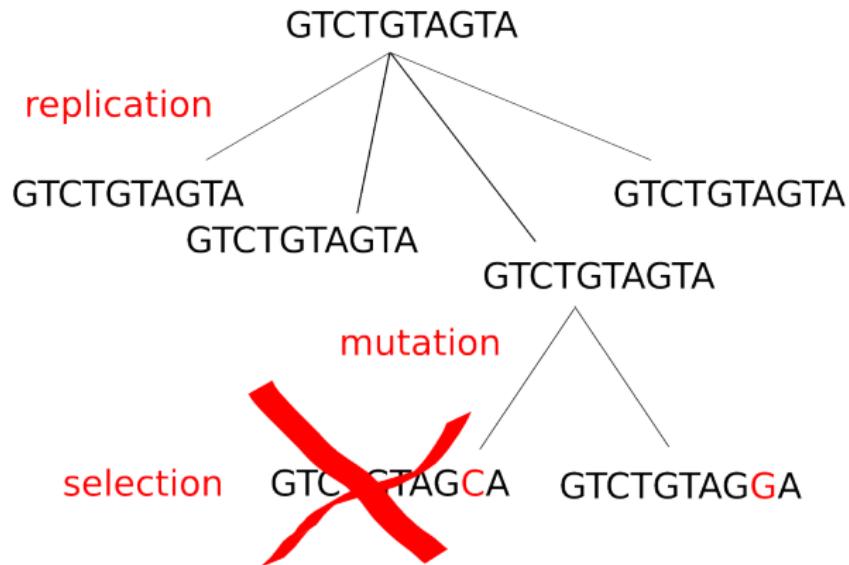


image (c) BW

- How far in evolution are sequences we can observe in different living species?
- More formally: Can we define a measure of sequence similarity

$$d : \Sigma^* \times \Sigma^* \rightarrow \mathcal{R}^+$$

approximating the true evolutionary distance?

- Hint: We should count the number of mutations leading to the observed divergence.

Problems with DNA evolution models

- Mutations occur on DNA level, but selection acts much higher: on the phenotype level.
- This makes the assumption of base independence invalid
- Long evolutionary times violate time-reversibility
- Multiplicative measure not too convenient in practice
- We can only account for substitutions, not for insertions or deletions

Suggested solutions:

- Use protein sequences for comparisons
- Define additive substitution matrices

Protein substitution matrices

- We are still assuming time-reversible Markov chain, but now in space of protein sequences.
- Matrix entries contain log-probabilities, leading to additive measures of similarity
- PAM (Point accepted mutations) matrices (Dayhoff, 1978) describe observed probabilities of occurrence of point mutations for a given average divergence (PAM1 = one mutation/100 bases, mostly used PAM250)
- BLOSUM (BLOCKS Substitution Matrix) (Henikoff, Henikoff 1992) were constructed using short protein alignments (Blocks) of given sequence identity.
e.g. BLOSUM80 was derived from sequences of $\geq 80\%$ identity

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3
S	1	-1	1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2		
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

Log-odds $\log \frac{P_{x,y}}{Q_x Q_y}$ instead of probabilities.

image (c) Durbin et al.

Quiz – using silent mutations

We know two types of mutations in DNA *silent* and *coding*

- Which of them are more interesting for calculating divergence between species?
- And which are more interesting for paternity testing?

Counting mutations – Hamming distance

- Hamming distance: a metric originating from Information theory
- Given two vectors of the same length, it returns the number of positions where they differ.
- $$D_H(s_1, s_2) = \sum_{i=1}^n \{1 : s_1[i] \neq s_2[i]; 0 : \text{otherwise}\}$$
- A proper distance (satisfies triangle inequality)

Insertions/deletions – small and large

- DNA polymerase can (rarely) slide over nucleotides
- especially over stretches of low complexity
- this leads to short deletions of DNA after replication
- Transposable elements lead to insertions of larger segments
- Chromosome recombination leads to duplications and deletions on different chromosomes at the same time

Insertions/deletions – symmetric cases

Number of mutations needed to evolve two sequences from a common ancestor is the same (under parsimony assumption) as the number of mutations needed to evolve one into the other

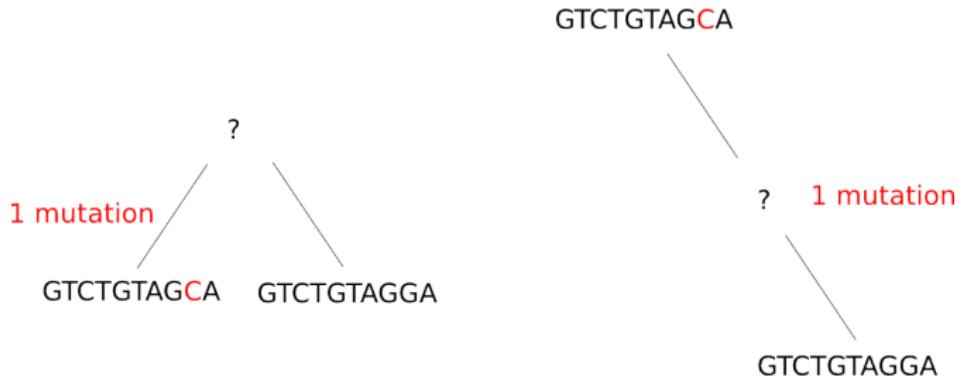


image (c) BW

Genes – units of evolutionary information

- Classically, genes are the **basic units of heritability**
- Georg Mendel (19th century) laid foundations of genetics with his experiments on peas
- He also introduced the term allele and formulated laws of inheritance (segregation and independence)
- He knew nothing about DNA!

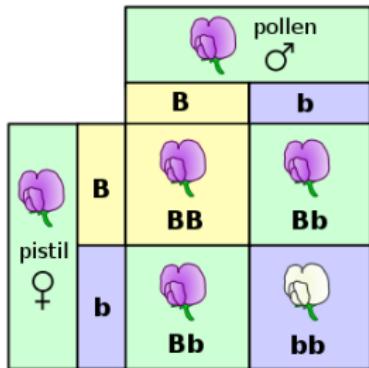


image (c) albiflora.eu

- Currently, we know that genes are carried by DNA
- Current definition of a gene is substantially more complex:
- *a locatable region of genomic sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions, and or other functional sequence regions* (Pearson, Nature, 2006)
- This is overly complex for our purposes, so
- We will be most concerned with *protein coding genes*

Edit distance – solution or a problem?

- We can introduce *edit distance*: the number of editing operations needed to transform one sequence into the other. These operations are:
 - Substitutions
 - Insertions
 - Deletions
- The *procedural* definition of the distance makes it difficult to work with
- Does it matter in what order I make the operations (*If I delete a character, I cannot substitute it anymore...*)
- It turns out the *optimal* edit distances are simpler and can be described in a formal way as sequence *alignments*

Sequence alignment – problem statement

For a given sequences s, t over an alphabet Σ , their alignment is a pair of words s', t' over the extended alphabet $\Sigma' = \Sigma \cup \{-\}$. Sequences s', t' need to satisfy the following:

- $|s'| = |t'|$
- $s'|_{\Sigma} = s$ and $t'|_{\Sigma} = t$
- for no position i , $s'[i] = t'[i] = -$

For example, one of the words HEAGAWGHEE and PAWHEAE is

HEAGAWGHE-E

--P-AW-HEAE

Number of possible alignments for words of length n

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

Scoring alignments: binary dotplots

Dotplot of the alignment of human haemoglobin α vs β chains

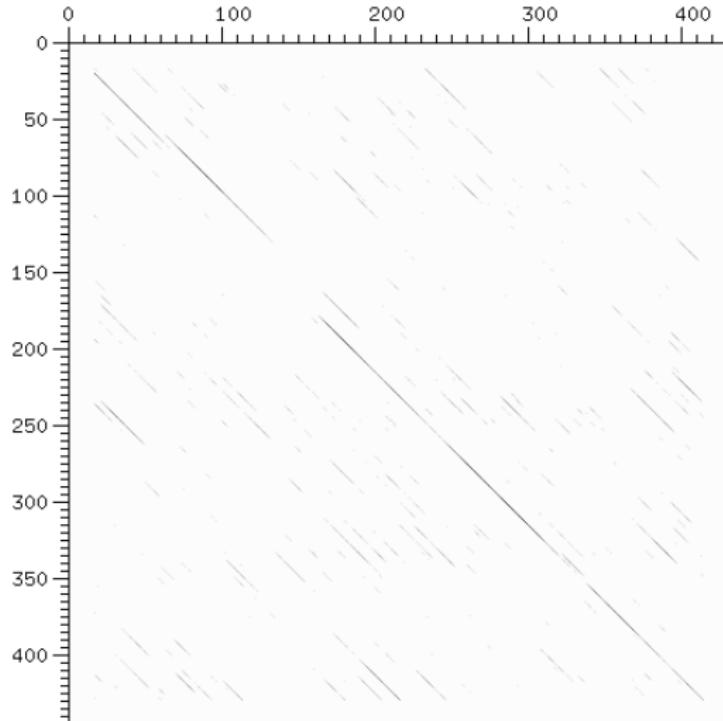


image (c) Hannes Luz

Scoring alignments: BLOSUM score matrix

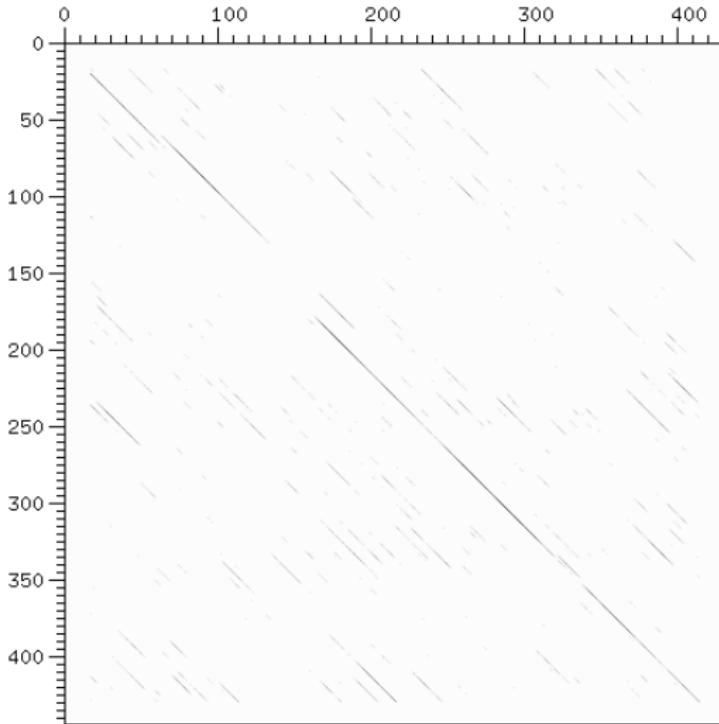


image (c) Hannes Luz

Recursive equation for sequence alignment

	H	E	A	G	A	W	G	H	E	E
P	-2	-1	-1	-2	-1	-4	-2	-2	-1	-1
A	-2	-1	5	0	5	-3	0	-2	-1	-1
W	-3	-3	-3	-3	-3	15	-3	-3	-3	-3
H	10	0	-2	-2	-2	-3	-2	10	0	0
E	0	6	-1	-3	-1	-3	-3	0	6	6
A	-2	-1	5	0	5	-3	0	-2	-1	-1
E	0	6	-1	-3	-1	-3	-3	0	6	6

image (c) Durbin et al.

Filling in the alignment matrix

$$F(i, j) = \max \begin{cases} F(i - 1, j - 1) + s(x_i, y_j), \\ F(i - 1, j) - d, \\ F(i, j - 1) - d. \end{cases}$$

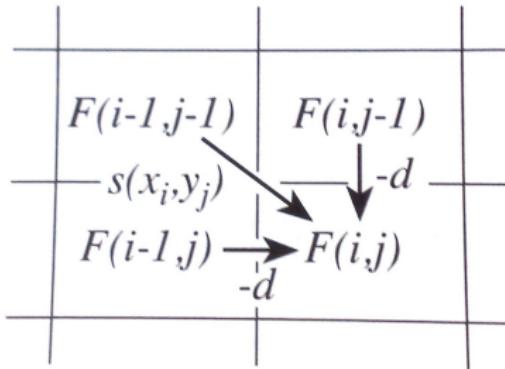


image (c) Durbin et al.

Tracing back alignments

	H	E	A	G	A	W	G	H	E	E
P	0 ← -8 ← -16 ← -24 ← -32 ← -40 ← -48 ← -56 ← -64 ← -72 ← -80									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
A	-8 -2 -9 -17 ← -25									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
W	-16 -10 -3 -4 ← -12 -20 ← -28 ← -36 ← -44 ← -52 ← -60									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
H	-24 -18 -11 -6 -7 -15 -5 ← -13 ← -21 ← -29 ← -37									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
E	-32 -14 -18 -13 -8 -9 -13 -7 -3 -11 -19									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
A	-40 -22 -8 ← -16 -16 -9 -12 -15 -7 3 -5									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
E	-48 -30 -16 -3 ← -11 -11 -12 -12 -15 -5 2									
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗

HEAGAWGHE-E

--P-AW-HEAE

image (c) Durbin et al.

Finding local alignments - Smith,Waterman '82

$$F(i, j) = \max \begin{cases} 0, \\ F(i-1, j-1) + s(x_i, y_j), \\ F(i-1, j) - d, \\ F(i, j-1) - d. \end{cases}$$

	H	E	A	G	A	W	G	H	E	E
P	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	5	0	0	0	0
W	0	0	0	0	2	0	20 ← 12 ← 4	0	0	0
H	0	10 ← 2	0	0	0	0	12 ↑ 18 ↑ 22 ← 14 ← 6	0	0	0
E	0	2	16 ← 8	0	0	4	10 ↑ 18 ↑ 28	20	0	0
A	0	0	8	21 ← 13	5	0	4 ↑ 10 ↑ 20	27	0	0
E	0	0	6	13	18	12 ← 4	0 ↑ 4 ↑ 16	26	0	0

AWGHE

AW-HE

image (c) Durbin et al.

Scoring alignments: general gap penalty

General gap penalty

$$F(i, j) = \max \begin{cases} F(i - 1, j - 1) + s(x_i, y_j), \\ F(k, j) + \gamma(i - k), & k = 0, \dots, i - 1, \\ F(i, k) + \gamma(j - k), & k = 0, \dots, j - 1. \end{cases}$$

Affine gap penalty (caching)

$$M(i, j) = \max \begin{cases} M(i - 1, j - 1) + s(x_i, y_j), \\ I_x(i - 1, j - 1) + s(x_i, y_j), \\ I_y(i - 1, j - 1) + s(x_i, y_j); \end{cases}$$

$$I_x(i, j) = \max \begin{cases} M(i - 1, j) - d, \\ I_x(i - 1, j) - e; \end{cases}$$

$$I_y(i, j) = \max \begin{cases} M(i, j - 1) - d, \\ I_y(i, j - 1) - e. \end{cases}$$

image (c) Durbin et al.