

Znajdowanie sekwencji w bazach danych

Bartek Wilczyński

16. kwietnia 2019

- Materiały do dzisiejszego wykładu:
<http://www.mimuw.edu.pl/~dojer/wobm/alg-heuryst.pdf>

How much similarity is there?

Znajdowanie sekwencji w bazach danych

Bartek Wilczyński

Organizacyjne

Why search for sequences

Approximate and heuristic searching

Many important genes are conserved between distantly related species

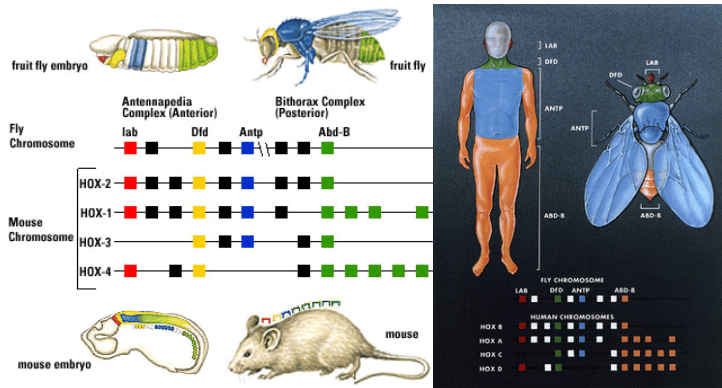


image sources biologycorner.org and ilbiologista.blogspot.com

Sequencing efforts are getting cheaper...

Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

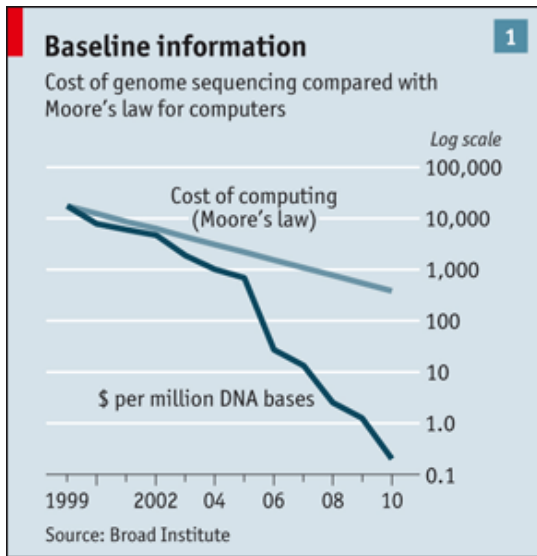


image source the economist

...and grow exponentially...

Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

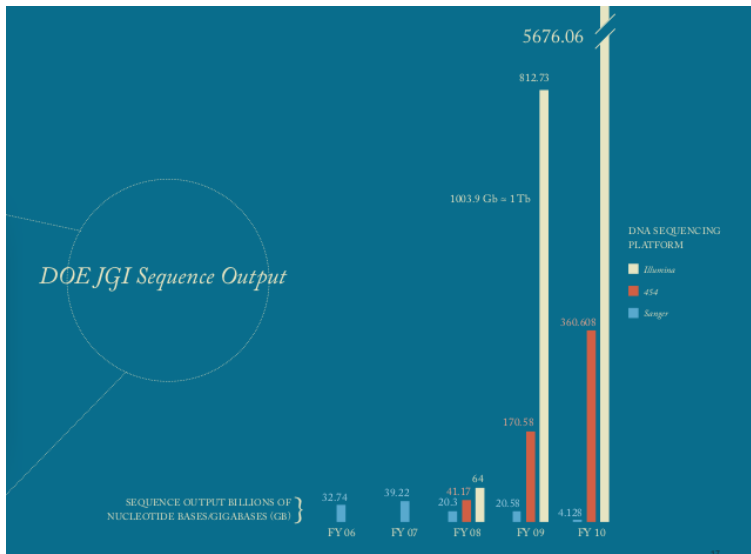


image source JGI annual report 2010

...and so do the databases of known sequences

Znajdowanie
sekwencji w
bazach
danych

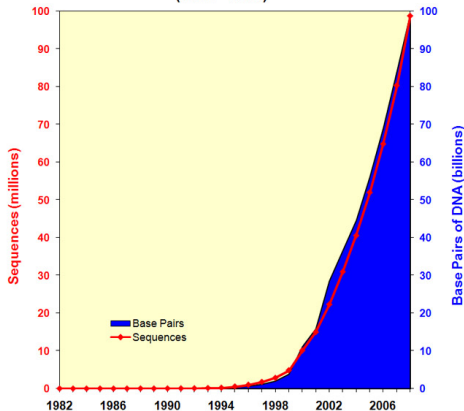
Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

Growth of GenBank
(1982 - 2008)



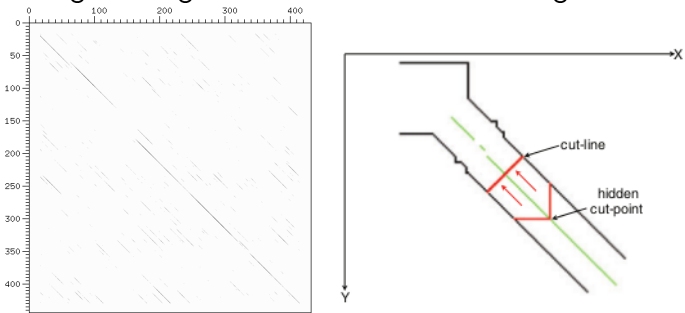
Genbank non-redundant nucleotide count is now $\geq 10^{11}$ and
sequence count $\geq 10^8$. image source NIH NCBI release notes

- Indeed, we can find similar sequences by comparing them with local sequence alignment methods
- Such algorithms run in $\mathcal{O}(n \cdot m)$ time scale
- How much would a Smith-Waterman analysis of a single new sequence (1000bp) against genbank take?
- How long for a genome with 10 thousand genes?
- How long for the JGI annual throughput?
- Can we wait that long?
- Can it be done faster?
- What assumptions do we need to make?

How to define our problem?

- We are looking only for **similar** sequences in the database, so most of our work with S-W algorithm is comparing sequences which will not show up in the result
- Can we tell if a sequence is *not-similar* more quickly than S-W?
- We need to define a meaningful way of specifying our definition of *not-similar*
- **We need an algorithm that can reject bad alignments based on a meaningful and computable criteria**

Good global alignments reside close to the diagonal



- Restricting to search within fixed distance from diagonal brings our computing time to almost linear
- but **not for local alignments**

image source: pecan algorithm

Second idea: FASTA matching short exact matches

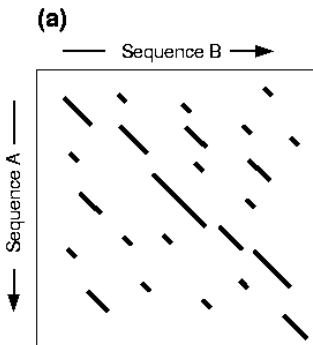
Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

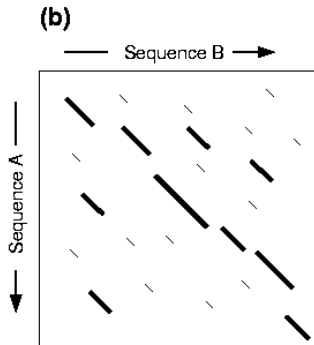
Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

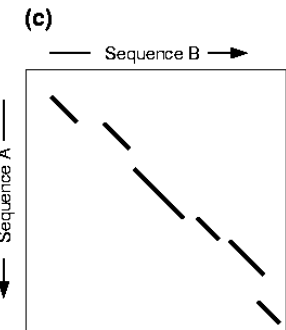


Find runs of identities

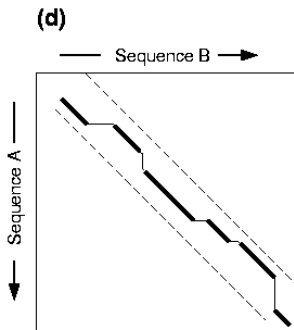


Re-score using PAM matrix
Keep top scoring segments.

image source GJ Barton



Apply "joining threshold"
to eliminate segments that
are unlikely to be part of the alignment
that includes highest scoring segment.



Use dynamic programming
to optimise the alignment in a
narrow band that encompasses
the top scoring segments.

Fourth idea: BLAST *hashing* words *similar* to query

Query sequence: PQGEFG

Word 1: PQQ

Word 2: QGE

Word 3: GEF

Word 4: EFG

Idea 3': BLAST extending words to segments

Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

Query sequence: R P P Q G L F

Database sequence: D P P E G V V

↳ Exact match is scanned.

Score: -2 7 7 2 6 1 -1

↳ HSP

Optimal accumulated score = $7+7+2+6+1 = 23$

image source wikipedia

Idea 3'': BLAST high scoring segment pairs (HSP)

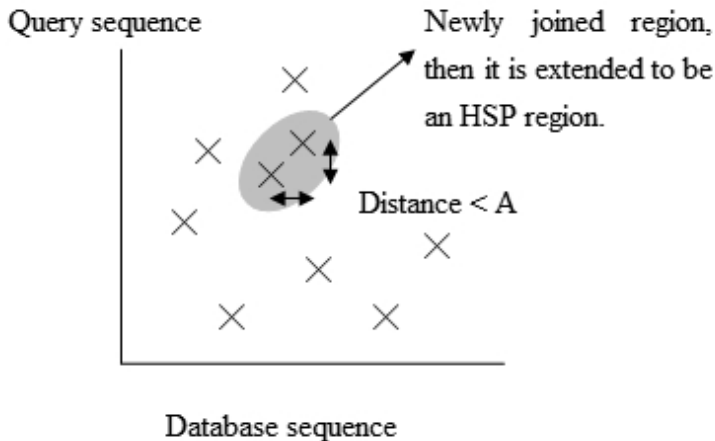


image source wikipedia

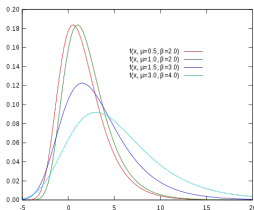
Idea 5: BLAST computing significance of an HSP

- Assume that you found an HSP, is it worth keeping it in the result?
- Behave like a collector: it's only worth keeping if it is rare
- Formally, we want matches which are **unlikely to occur by random** in similar situations (defined by size and composition of the query and database)
- In statistics, we are performing **hypothesis testing**: under **null hypothesis**, there are no matching sequences in the database
- We are interested in the probability of observing a given score (or higher) under assumption of the null model

Idea 5: BLAST computing significance of an HSP

- We cannot really estimate this probability by Monte-Carlo (data is too large for large-scale sampling)
- It is assumed, that it should follow the extreme value distribution (Gumbel distribution)

$$p(s \geq x) = 1 - \exp(-e^{-\lambda(x-\mu)}), \mu = \frac{\log(Km'n')}{\lambda}$$



parameters K and λ can be estimated from data, then the E-value is computed $E = pD$, where D is the number of sequences in the database (similar to Bonferroni correction)

Ψ -BLAST - position-specific-iterative BLAST

Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

- Since we are using short words that are similar to the query to find hits, we could use profiles (or position-specific score matrices) to generate such words
- Psi-BLAST (Ψ -BLAST) uses this idea to iteratively search for distant relatives of the query sequence
- The main idea is that we can create a i -th MSA (multiple sequence alignment) of the BLAST hits of the i -th iteration, and then generate new short words for the $(i+1)$ -th iteration from the profile of this i -th MSA.
- Additionally, Ψ -BLAST modifies the default significance thresholds - we are searching for *distant* relatives after all
- Since we use lowered thresholds, we also need to improve the statistics - now K and λ are estimated depending on the local aminoacid frequencies

Ψ -BLAST - position-specific-iterative BLAST

Znajdowanie
sekwencji w
bazach
danych

Bartek
Wilczyński

Organizacyjne

Why search
for sequences

Approximate
and heuristic
searching

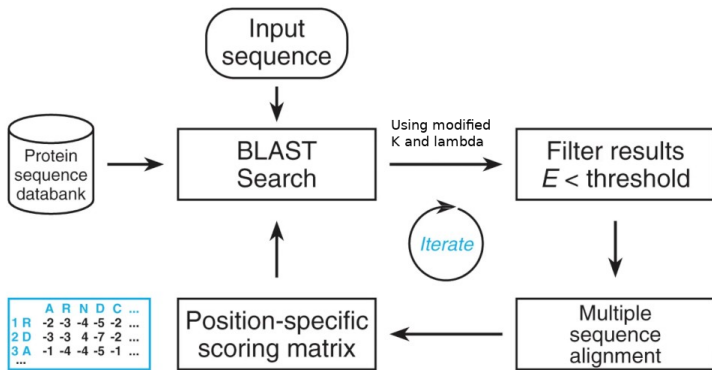


image modified from <https://www.youtube.com/watch?v=HIDcPyX4YZg>

- Very fast algorithm
- Rather complicated heuristic approach
- Many very specialized variants (blastn, blastp, blastx, etc.)
- Does not attempt to find a global alignment, but rather generate a number of *significant* predictions
- Computing e-values and bit-scores (e-value normalized for m and n) is a very important feature