## Cap. 6 How Do We Compare Biological Sequences

- From Sequence Comparison to Biological Insights
- The Alignment Game and the Longest Common Subsequence
- The Manhattan Tourist Problem
- Dynamic Programming and Backtracking Pointers
- From Manhattan to an Arbitrary Directed Acyclic Graph
- From Global to Local Alignment [Matrici di Score…]
- Penalizing Insertions and Deletions in Sequence Alignment
- Space-Efficient Sequence Alignment
- Multiple Sequence Alignment OGGI

1

## EMBL-EBI

| Type | REST Service | Description |
|---|---|---|
| Global Alignment | EMBOSS needle | Needleman-Wunsch global alignment using EMBOSS needle. |
| | EMBOSS stretcher | Myers and Miller global alignment using EMBOSS stretcher. `linear space` |
| | GGSEARCH2SEQ | Needleman-Wunsch global alignment using GGSEARCH2SEQ. `finds internal duplications` |
| Local Alignment | EMBOSS water | Smith-Waterman local alignment using EMBOSS water. |
| | EMBOSS matcher | Waterman-Eggert local alignment using EMBOSS matcher. |
| | LALIGN | Huang and Miller *sim* local alignment using lalign. |
| | SSEARCH2SEQ | Smith-Waterman local alignment using SSEARCH2SEQ. |
| Genomic Alignment | GeneWise | Comparing a protein sequence to a genomic DNA sequence. |

https://www.ebi.ac.uk/jdispatcher/

2

## Sequence Similarity Search

Find sequences in databases based on similarity.

euristiche

NCBI BLAST | PSI-BLAST | FASTA | SSEARCH | PSI-Search | PSI-Search2 | GGSEARCH | GLSEARCH | FASTM/S/F | Less

3

## FASTA and BLAST

**FASTA** Pearson, W.R. & Lipman, D.J. (1988) Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. USA 85:2444-2448.

https://www.ebi.ac.uk/Tools/sss/fasta/nucleotide.html

**BLAST**

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410.

https://blast.ncbi.nlm.nih.gov/Blast.cgi

4

## FASTA - summary

FASTA (pronounced "fast A»)
"FAST-All", because it works with any alphabet, and it is an extension of "FAST-P" (protein) and "FAST-N" (nucleotide) alignment.

**Come possiamo giudicare «buono» un allineamento?**
E' tanto migliore quanto è più improbabile che sia frutto del caso, cioè migliore rispetto ad un allineamento tra due sequenze di caratteri casuali della stessa lunghezza.

5

## FASTA - summary

The FASTA algorithm is a heuristic method for string comparison.
The **Z-score** (comparison of an actual alignment score with the scores obtained on a set of random sequences) certify how meaningful is the obtained comparison (number of standard deviations separating the query score (opt) from the mean of the random scores).

**Come possiamo giudicare «buono» un allineamento?**
E' tanto migliore quanto è più improbabile che sia frutto del caso, cioè migliore rispetto ad un allineamento tra due sequenze di caratteri casuali della stessa lunghezza.

6

# FASTA - summary

The FASTA algorithm is a heuristic method for string comparison.
The **Z-score** (comparison of an actual alignment score with the scores obtained on a set of random sequences) certify how meaningful is the obtained comparison (number of standard deviations separating the query score (opt) from the mean of the random scores).

It performs Local alignment of sequences.

The heuristic method uses hash coding in which the sequence is broken into small "words or K-tuples"of specific sizes (k=6 for nucleotide seq.)

Dynamic programming methods consume time to analyse and search the entire database. *Heuristic approach overcomes this drawback. FASTA focused on the diagonal of the PD matrix.*

7

# FASTA – Step 1: offset definition

- For each sequence (A) in the database, a table is created by considering the position of the amino acids taken individually (ktup=1) or as pairs (ktup=2); nucleotide sequences ktup is 4 or 6.

| Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Sequence A | F | L | W | R | T | W | S |

| | |
|---|---|
| F | 1 |
| L | 2 |
| W | 3, 6 |
| R | 4 |
| T | 5 |
| S | 7 |

| Position | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Sequence B | S | W | R | T | W | T |

| | |
|---|---|
| S | 1 |
| W | 2, 5 |
| R | 3 |
| T | 4, 6 |

The same is done for the query sequence B

8

# FASTA – Step 1: offset definition

- Local similarity: offsets
- Insertions / deletions that are forbidden.
- The 10 best regions of similarity are "stored" for further analysis.

| | DELTA | OFFSET |
|---|---|---|
| S | 7 - 1 | 6 |
| W | 3 – 2 | 1 |
| W | 3 – 5 | -2 |
| W | 6 – 2 | 4 |
| W | 6 – 5 | 1 |
| R | 4 – 3 | 1 |
| T | 5 – 4 | 1 |
| T | 5 – 6 | -1 |

| Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Sequence A | F | L | W | R | T | W | S |

| | |
|---|---|
| F | 1 |
| L | 2 |
| W | 3, 6 |
| R | 4 |
| T | 5 |
| S | 7 |

| Position | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Sequence B | S | W | R | T | W | T |

| | |
|---|---|
| S | 1 |
| W | 2, 5 |
| R | 3 |
| T | 4, 6 |

9

# FASTA – Step 1: offset definition

- Local similarity: offsets
- Insertions / deletions that are forbidden.
- The 10 best regions of similarity are "stored" for further analysis.

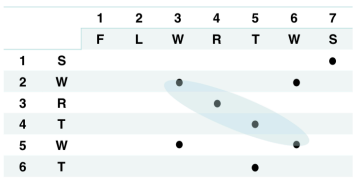
| | DELTA | OFFSET |
|---|---|---|
| S | 7 - 1 | 6 |
| W | 3 – 2 | 1 |
| W | 3 – 5 | -2 |
| W | 6 – 2 | 4 |
| W | 6 – 5 | 1 |
| R | 4 – 3 | 1 |
| T | 5 – 4 | 1 |
| T | 5 – 6 | -1 |

| Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Sequence A | F | L | W | R | T | W | S |

| | |
|---|---|
| F | 1 |
| L | 2 |
| W | 3, 6 |
| R | 4 |
| T | 5 |
| S | 7 |

| Position | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Sequence B | S | W | R | T | W | T |

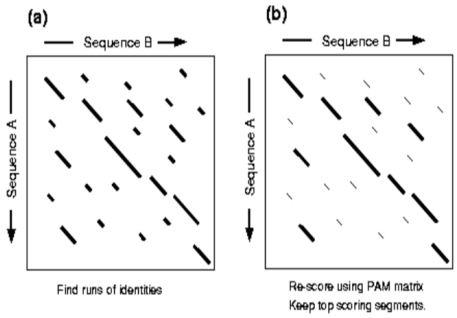
| | |
|---|---|
| S | 1 |
| W | 2, 5 |
| R | 3 |
| T | 4, 6 |




10

## FASTA – Step 2: Evaluation of Amino Acids Substitution

- This step evaluates any *replacements* that occurred between amino acids in the 10 best regions of similarity selected in the first phase, using a scoring matrix, e.g., PAM250 [INIT1]




11

## FASTA – Step 3: Join several initial regions

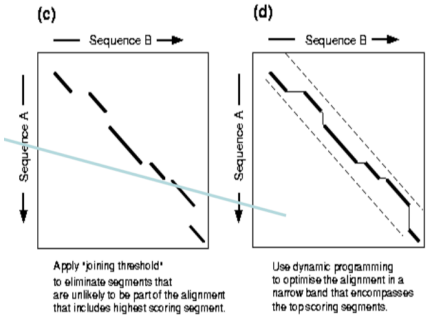
- The initial score INIT1 is recalculated being redefined on the basis of created joins, and that is called INITN.

FASTA carries out an assessment about possible ways to connect (join) different starting regions.

-No overlapping between regions

-Score must be higher than a "threshold value»

-Introduction of a penalty score (-16) for each gap (the gap penalty).

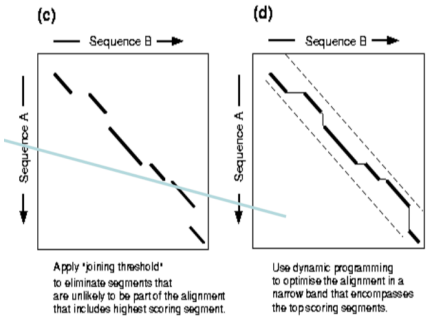



12

## FASTA – Step 4: Deletions and insertions evaluation

- Sequences showing greater similarity are aligned to the input sequence using a modified version of Needleman and Wunsch' algorithm.

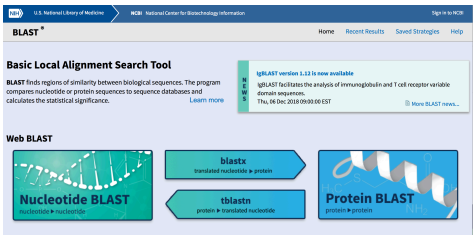
OPT: optimized score, after evaluation of the insertions and deletions




13

## What is BLAST?

- Freely available tool compares a protein or DNA sequence to other sequences in various databases
- Helps researchers in identifying sequences similar to the query sequence
- Helps to infer *homology* (statistically significant similarity that reflects common ancestry ) and identify sequences that may be important for function
- https://blast.ncbi.nlm.nih.gov/Blast.cgi

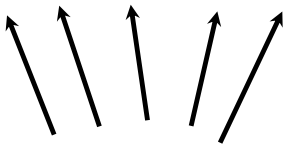



14

## Phase 1: Compile a List of Words

**Example:** for a protein query …FSGTWYA…
A list of words from the query sequence (w=3) is:

FSG SGT GTW TWY WYA



Query Words

For the query word …FSGTWYA… (in red)
A list of neighborhood words is generated:
GTW (all three the same)
GSW, ATW, NTW, GTY, GNW, etc. (*two of the three the same*)

15

## Phase 1: Compile a List of Words

For the query word …FSGTWYA… (in red)

Neighborhood words are given a score based on their similarity to the query word

*exact match results in the highest score

neighborhood word hits > threshold (T=11)

GTW 6+5+11 = 22*
GSW 6+1+11 = 18
ATW 0+5+11 = 16
NTW 0+5+11 = 16
GTY 6+5+2 = 13

~~GNW 10~~
~~GAW 9~~

neighborhood word hits below the threshold are eliminated

16

## Phase 1: Compile a List of Words

The same analysis is then done for all the query words of a certain length (here a length of 3) generated from the …FSGTWYA… sequence

i.e. … FSG SGT GTW TWY WYA…

17

## Phase 2: Scan the Database

Each database sequence (*subject*) is scanned for the words from the list

Query: FSGTWYA
Subject: DMGSWHK

List of neighborhood words:
GTW, GSW, ATW, NTW, GTY, GNW

18

## Phase 3: Extend to Find High Scoring Pairs

KENFDKARFSGTWYAMAKKDPEG 50 RBP (query)
MKGLDIQKVAGSWYSLAMAASD 44 lactoglobulin (hit)

Extend Extend

Match between the query and the sequence found

The search is extended in either direction to identify high scoring segment pairs (HSPs)

approssimate dynamic programming

HSPs are a pair of alignments for which the similarity scores meets or exceeds a threshold

seed and extend technique

19

- Mega BLAST uses the greedy algorithm for nucleotide sequence alignment search.

**High speed BLASTN: an accelerated MegaBLAST search tool**

Ying Chen [1], Weicai Ye [1], Yongdong Zhang [2], Yuesheng Xu [3]

Affiliations + expand

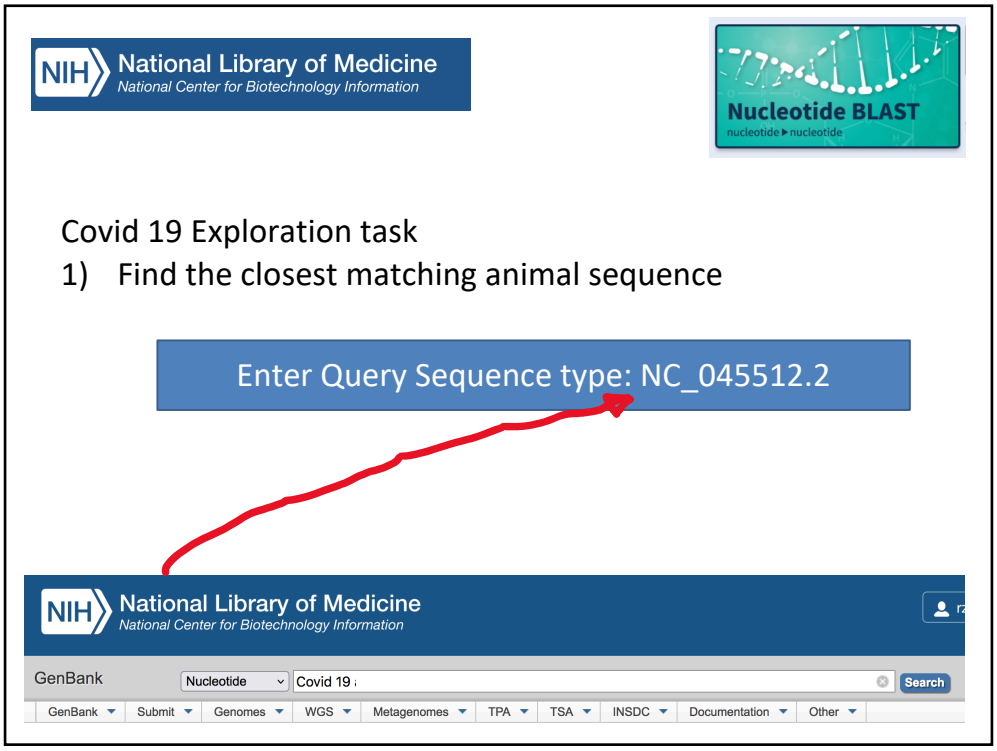
PMID: 26250111 PMCID: PMC4652774 DOI: 10.1093/nar/gkv784

Free PMC article

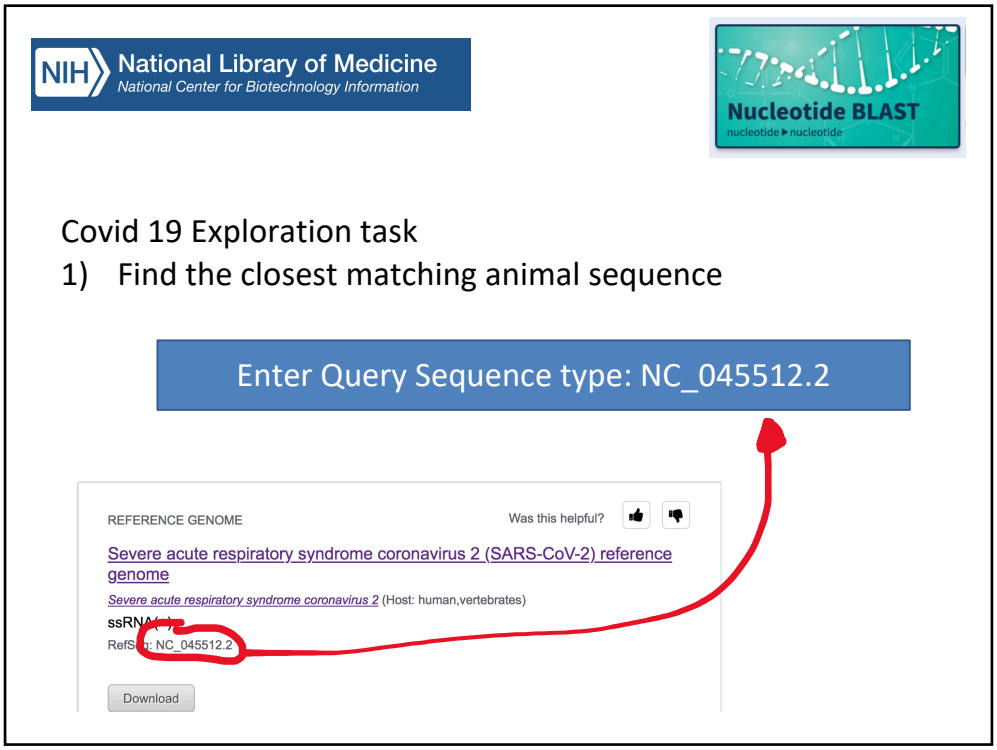
**Abstract**

Sequence alignment is a long standing problem in bioinformatics. The Basic Local Alignment Search Tool (BLAST) is one of the most popular and fundamental alignment tools. The explosive growth of biological sequences calls for speedup of sequence alignment tools such as BLAST. To this end, we develop high speed BLASTN (HS-BLASTN), a parallel and fast nucleotide database search tool that accelerates MegaBLAST--the default module of NCBI-BLASTN. HS-BLASTN builds a new lookup table using the FMD-index of the database and employs an accurate and effective seeding method to find short stretches of identities (called seeds) between the query and the database. HS-BLASTN produces the same alignment results as MegaBLAST and its computational speed is much faster than MegaBLAST. Specifically, our experiments conducted on a 12-core server show that HS-BLASTN can be 22 times faster than MegaBLAST and exhibits better parallel performance than MegaBLAST. HS-BLASTN is written in C++ and the related source code is available at https://github.com/chenying2016/queries under the GPLv3 license.

20


National Library of Medicine
National Center for Biotechnology Information
Nucleotide BLAST
nucleotide ▸ nucleotide
Covid 19 Exploration task
1) Find the closest matching animal sequence
Enter Query Sequence type: NC_045512.2
National Library of Medicine
National Center for Biotechnology Information
GenBank
Nucleotide
Covid 19
Search
GenBank
Submit
Genomes
WGS
Metagenomes
TPA
TSA
INSDC
Documentation
Other

21


National Library of Medicine
National Center for Biotechnology Information
Nucleotide BLAST
nucleotide ▸ nucleotide
Covid 19 Exploration task
1) Find the closest matching animal sequence
Enter Query Sequence type: NC_045512.2
REFERENCE GENOME
Was this helpful?
Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) reference genome
Severe acute respiratory syndrome coronavirus 2 (Host: human,vertebrates)
ssRNA
RefSeq: NC_045512.2
Download

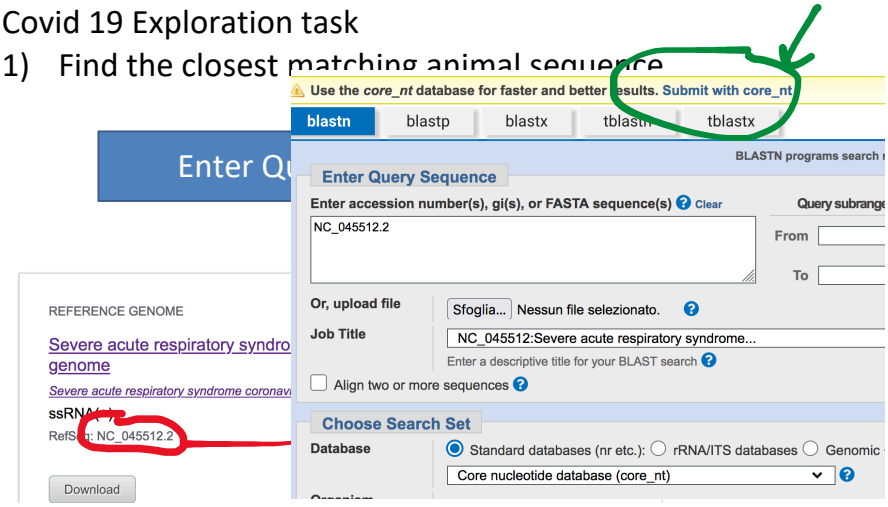
22

NIH National Library of Medicine
National Center for Biotechnology Information

Nucleotide BLAST
nucleotide ▸ nucleotide

Covid 19 Exploration task

1) Find the closest matching animal sequence




23

NIH National Library of Medicine
National Center for Biotechnology Information

Nucleotide BLAST
nucleotide ▸ nucleotide

Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of **COVID19**, **SARS**, Bat SARS Coronavirus Rf1 and BAT Coronavirus

Enter Query Sequence 1 type: NC_045512.2

Enter Query Sequence 2 type: NC_004718

CHECK: Align two or more sequences

SEE «Alignments»

24

## Assignment

- Rifare quanto fatto in aula sul confronto tra Covid 19 e le altre 2 sequenze, come indicato nelle slide successive.
- Sarà caricato un file pdf con la specifica sequenza di azioni che occorre fare e cosa cercare.

25

NIH National Library of Medicine
National Center for Biotechnology Information

Nucleotide BLAST
nucleotide ▸ nucleotide

Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of **COVID19**, SARS, **Bat SARS Coronavirus Rf1** and BAT Coronavirus

Enter Query Sequence 1 type: NC_045512.2

Enter Query Sequence 3 type: DQ412042.1

CHECK: Align two or more sequences

SEE «Alignments»

26

NIH National Library of Medicine
National Center for Biotechnology Information


Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of **COVID19**, SARS, Bat SARS Coronavirus Rf1 and **BAT Coronavirus**

Enter Query Sequence 1 type: NC_045512.2

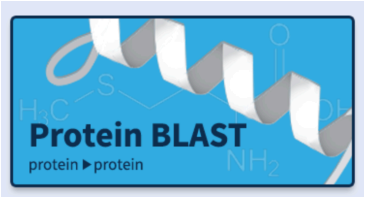
Enter Query Sequence 4 type: NC014470.1

CHECK:Align two or more sequences

SEE «Alignments»

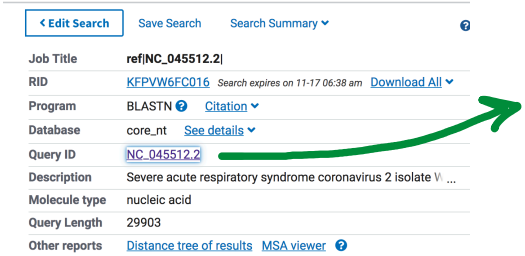
27

NIH National Library of Medicine
National Center for Biotechnology Information




Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of COVID19, SARS, Bat SARS Coronavirus Rf1 and BAT Coronavirus
3) Perform BLAST comparisons between Spyke protein of COVID19 and SARS




CLICCA QUI PER LA SEQUENZA

28

## Assignment

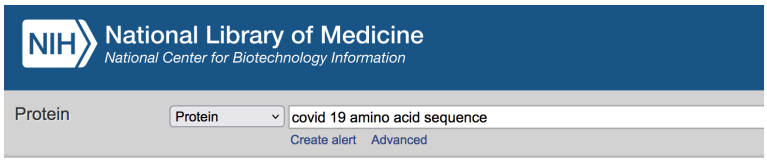
- Rifare quanto fatto in aula sulla proteina Spike, come indicato nella slide successiva.
- Il file pdf che sarà pubblicato conterrà anche le specifiche di questo task

29

NIH National Library of Medicine
National Center for Biotechnology Information


Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of COVID19, SARS, Bat SARS Coronavirus Rf1 and BAT Coronavirus
3) Perform BLAST comparisons between Spyke protein of COVID19 and SARS




poi cerca «CODICE DELLA protein S» nel file…

30

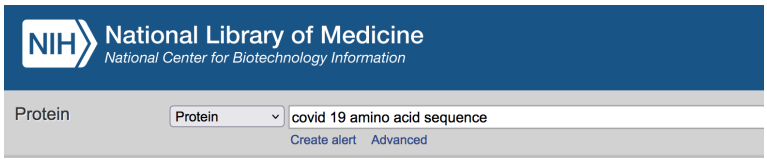
NIH National Library of Medicine
National Center for Biotechnology Information

Protein BLAST
protein ▶ protein

Covid 19 Exploration task

1) Find the closest matching animal sequence
2) Compare by BLAST the nucleotide sequences of COVID19, SARS, Bat SARS Coronavirus Rf1 and BAT Coronavirus
3) Perform BLAST comparisons between Spyke protein of COVID19 and SARS

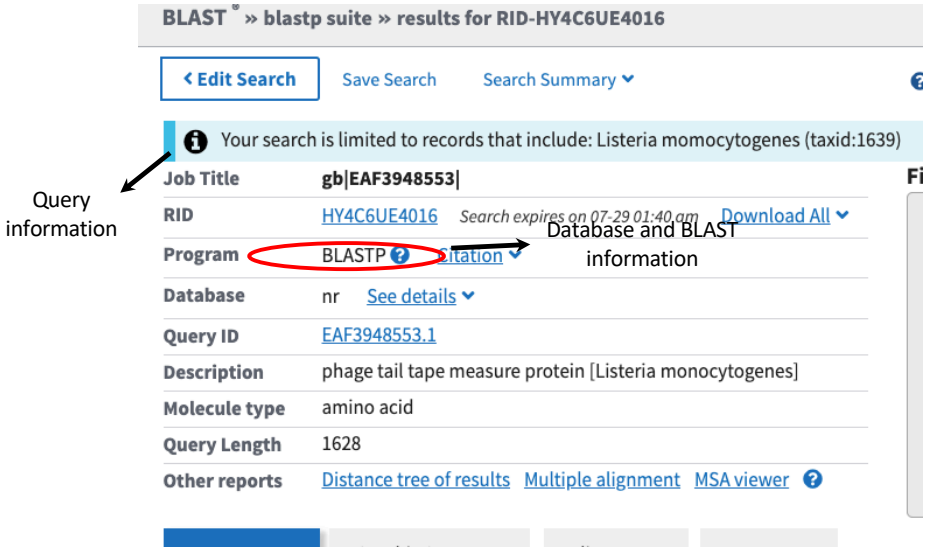
In the case of a protein alignments, identical matches are marked by the letter code and homologous matches by a '+' symbol in between the alignments




poi cerca «CODICE DELLA protein S» nel file…

31

## BLAST Results

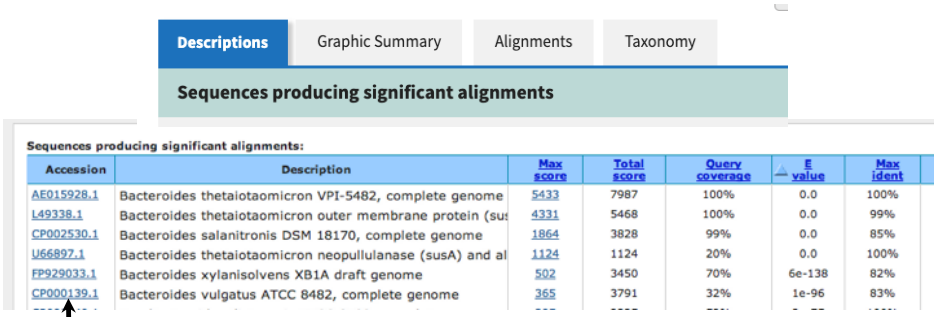



Query information

Database and BLAST information

32

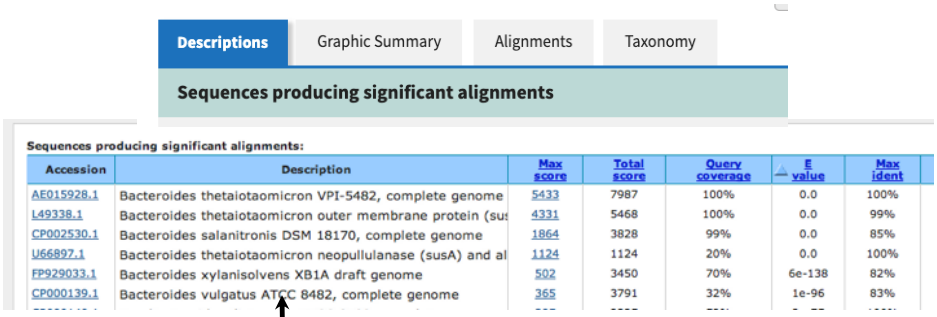
## BLAST Results – Accession Numbers




Accession Number: NCBI record

33

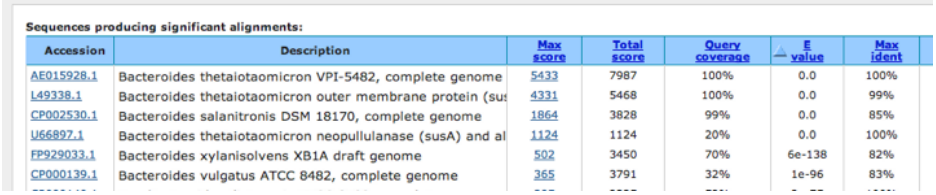
## BLAST Results – Accession Numbers




Sequence Description

34

## BLAST Results – Scores



Sequences producing significant alignments:

| Accession | Description | Max score | Total score | Query coverage | E value | Max ident |
|---|---|---|---|---|---|---|
| AE015928.1 | Bacteroides thetaiotaomicron VPI-5482, complete genome | 5433 | 7987 | 100% | 0.0 | 100% |
| L49338.1 | Bacteroides thetaiotaomicron outer membrane protein (sus | 4331 | 5468 | 100% | 0.0 | 99% |
| CP002530.1 | Bacteroides salanitronis DSM 18170, complete genome | 1864 | 3828 | 99% | 0.0 | 85% |
| U66897.1 | Bacteroides thetaiotaomicron neopullulanase (susA) and al | 1124 | 1124 | 20% | 0.0 | 100% |
| FP929033.1 | Bacteroides xylanisolvens XB1A draft genome | 502 | 3450 | 70% | 6e-138 | 82% |
| CP000139.1 | Bacteroides vulgatus ATCC 8482, complete genome | 365 | 3791 | 32% | 1e-96 | 83% |

Score: In context of an alignment, a score describes the overall quality of the alignment. Higher numbers correspond to higher similarity

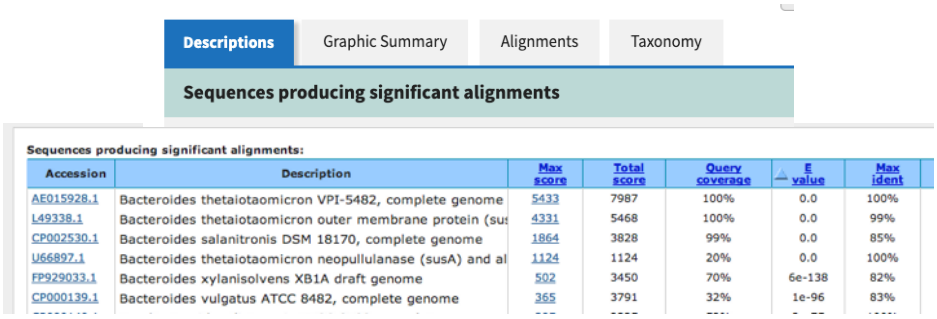
Raw score: is calculated from the substitution matrix and parameters used to assess the pair-wise alignments

Max score (bit score): is calculated from the raw score by normalizing with the statistical variables that define a given scoring system

Total score: includes score from non-contiguous portions of the subject sequence that match the query sequence

35

## BLAST Results – Query Coverage



Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments

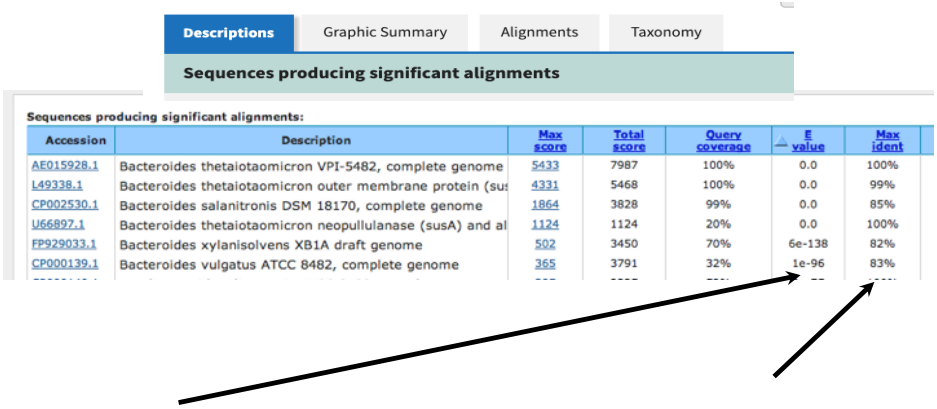
Sequences producing significant alignments:

| Accession | Description | Max score | Total score | Query coverage | E value | Max ident |
|---|---|---|---|---|---|---|
| AE015928.1 | Bacteroides thetaiotaomicron VPI-5482, complete genome | 5433 | 7987 | 100% | 0.0 | 100% |
| L49338.1 | Bacteroides thetaiotaomicron outer membrane protein (sus | 4331 | 5468 | 100% | 0.0 | 99% |
| CP002530.1 | Bacteroides salanitronis DSM 18170, complete genome | 1864 | 3828 | 99% | 0.0 | 85% |
| U66897.1 | Bacteroides thetaiotaomicron neopullulanase (susA) and al | 1124 | 1124 | 20% | 0.0 | 100% |
| FP929033.1 | Bacteroides xylanisolvens XB1A draft genome | 502 | 3450 | 70% | 6e-138 | 82% |
| CP000139.1 | Bacteroides vulgatus ATCC 8482, complete genome | 365 | 3791 | 32% | 1e-96 | 83% |

Fraction of the query sequence that matches the subject sequence

36

## BLAST Results – E-value and Maximum Identity

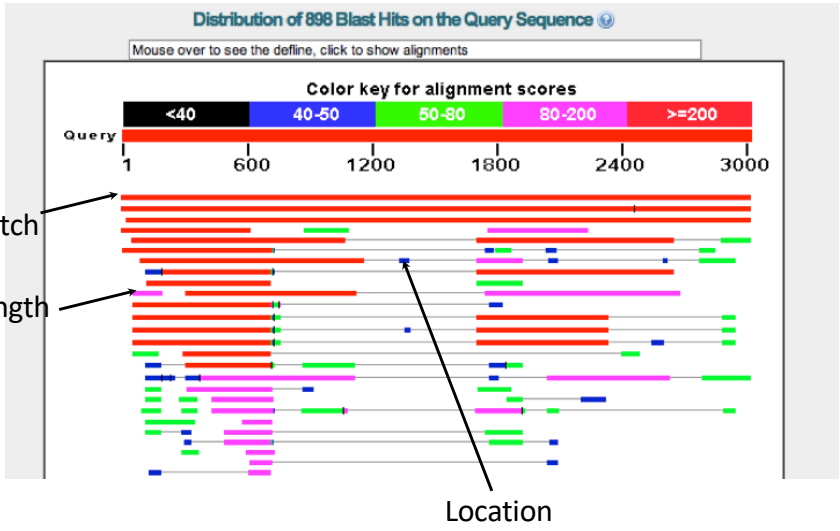



parameter that describes the number of hits one can "expect" to see by chance when searching a database of a particular size. *The lower the E-value, the more "significant" the match is.*

Match to the subject sequence with the higher percentage of identical bases

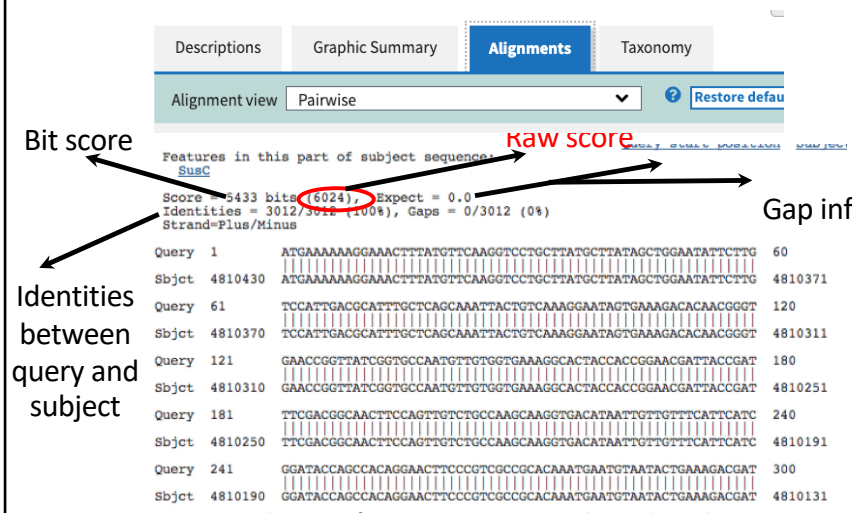
37

## BLAST Results




38

## BLAST Results – Alignment




Bit score

Raw score

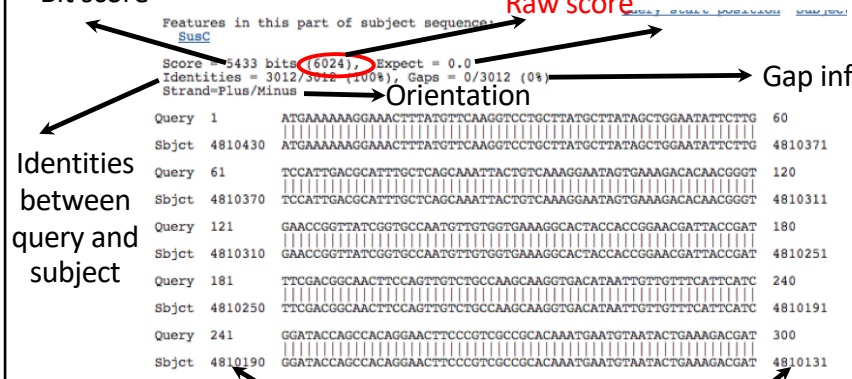
Gap information

Identities between query and subject

In the case of a protein alignments, identical matches are marked by the letter code and homologous matches by a '+' symbol in between the alignments

39

## BLAST Results – Alignment




Bit score

Raw score

Gap information

Orientation

Identities between query and subject

Co-ordinates for query and subject sequence

40

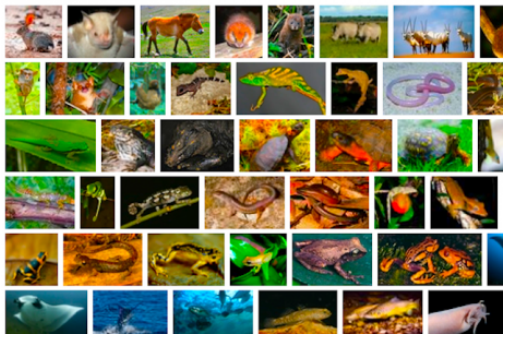
## How Do We Compare Biological Sequences

- From Sequence Comparison to Biological Insights
- The Alignment Game and the Longest Common Subsequence
- The Manhattan Tourist Problem
- Dynamic Programming and Backtracking Pointers
- From Manhattan to an Arbitrary Directed Acyclic Graph
- From Global to Local Alignment
- Penalizing Insertions and Deletions in Sequence Alignment
- Space-Efficient Sequence Alignment
- **Multiple Sequence Alignment**

41

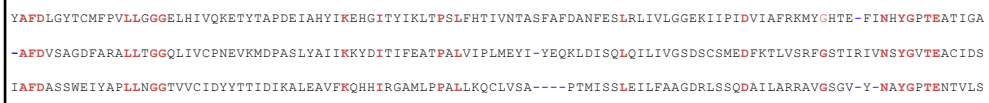
## From Pairwise to Multiple Alignment

- Up until now we have only tried to align two sequences.
- A faint (and statistically insignificant) similarity between two sequences becomes significant if it is present in many other sequences.
- Multiple alignments can reveal subtle similarities that pairwise alignments do not reveal.



42

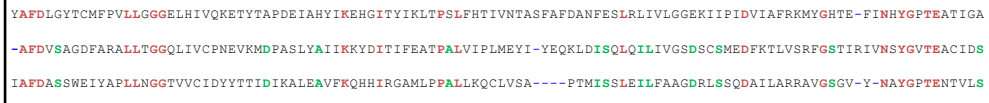
## Alignment of Three A-domains




We found 19 conservative columns…

43

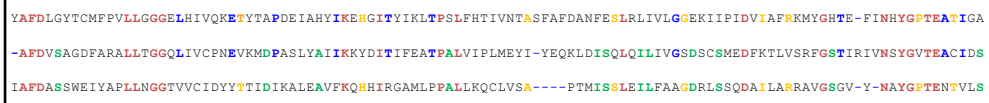
## Alignment of Three A-domains




We can find many pairwise similarities…

44

## Alignment of Three A-domains




but only if we compare all of them at once,
we find "that" similarity…

45

## Generalizing Pairwise to Multiple Alignment

- Alignment of 2 sequences is a 2-row matrix.
- Alignment of 3 sequences is a 3-row matrix

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

- Our scoring function should score alignments with conserved columns higher than other ones.

46

## Alignments = Paths in 3-D

47

## Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

| | | | | | |
|---|---|---|---|---|---|
| | A | -- | T | G | C |
| | A | A | T | -- | C |
| | -- | A | T | G | C |

48

# Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

| 0 | 1 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
|  | A | -- | T | G | C |
|  | A | A | T | -- | C |
|  | -- | A | T | G | C |

#symbols up to a given position

49

# Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

| 0 | 1 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
|  | A | -- | T | G | C |
| 0 | 1 | 2 | 3 | 3 | 4 |
|  | A | A | T | -- | C |
|  | -- | A | T | G | C |

#symbols up to a given position

50

# Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

| 0 | 1 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | A | -- | T | G | C |
| 0 | 1 | 2 | 3 | 3 | 4 |
| | A | A | T | -- | C |
| 0 | 0 | 1 | 2 | 3 | 4 |
| | -- | A | T | G | C |

Let's take the columns…

and as points in the three dimensional space.

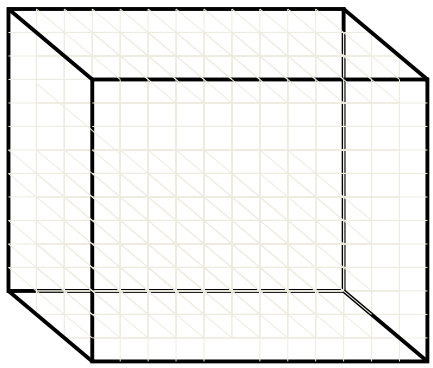
51

# Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

(0,0,0)→(1,1,0)→(1,2,1) →(2,3,2) →(3,3,3) →(4,4,4)

| 0 | 1 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | A | -- | T | G | C |
| 0 | 1 | 2 | 3 | 3 | 4 |
| | A | A | T | -- | C |
| 0 | 0 | 1 | 2 | 3 | 4 |
| | -- | A | T | G | C |



52

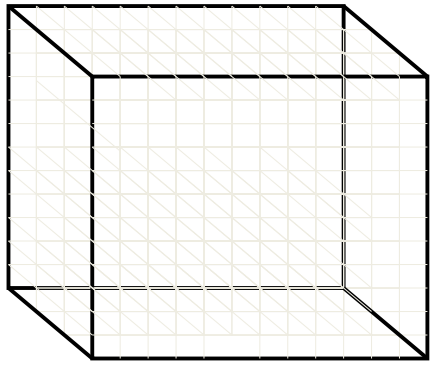
# Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

(0,0,0)→(1,1,0)→(1,2,1) →(2,3,2) →(3,3,3) →(4,4,4)

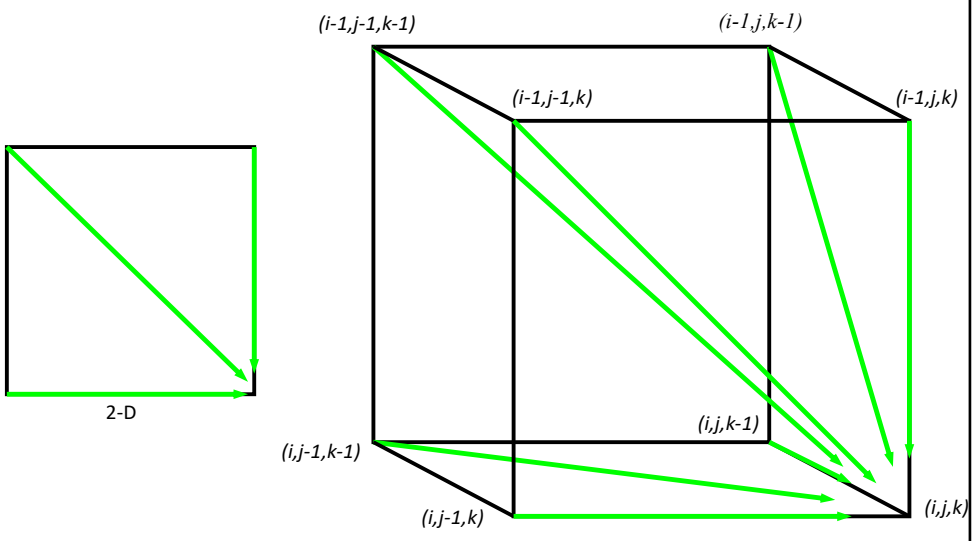
source and sink in our 3-dim. Manhattan grid

| 0 | 1 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | A | -- | T | G | C |
| 0 | 1 | 2 | 3 | 3 | 4 |
| | A | A | T | -- | C |
| 0 | 0 | 1 | 2 | 3 | 4 |
| | -- | A | T | G | C |



53

# 2-D Alignment Cell versus 3-D Alignment Cell




54

## Moving to Multiple Sequences

**Multiple Alignment Problem:** *Find the highest-scoring alignment between multiple strings.*

- **Input:** A collection of *t* strings (and some way of scoring columns of a multiple alignment).
- **Output:** A multiple alignment of these strings having maximum score.

55

## Moving to Multiple Sequences

**Multiple Alignment Problem:** *Find the highest-scoring alignment between multiple strings.*

- **Input:** A collection of *t* strings (and some way of scoring columns of a multiple alignment).
- **Output:** A multiple alignment of these strings having maximum score.

**STOP:** What algorithm would you propose to solve this problem?

56

## Multiple Alignment: Dynamic Programming

$$s_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) \\ s_{i-1,j-1,k} + \delta(v_i, w_j, -) \\ s_{i-1,j,k-1} + \delta(v_i, -, u_k) \\ s_{i,j-1,k-1} + \delta(-, w_j, u_k) \\ s_{i-1,j,k} + \delta(v_i, -, -) \\ s_{i,j-1,k} + \delta(-, w_j, -) \\ s_{i,j,k-1} + \delta(-, -, u_k) \end{cases}$$

Maximum among all 7 neighbors + score for moving

- $\delta(x, y, z)$ is an entry in the 3-D scoring matrix.

57

## Multiple Alignment: Running Time

- For 3 sequences of length $n$, the run time is proportional to $7n^3$ *(each of n cubed node has 7 neighbors to choose from).*
- For a $k$-way alignment, build a $k$-dimensional Manhattan graph with
  - $n^k$ nodes
  - most nodes have $2^k - 1$ incoming edges.
  - Runtime: $O(2^k n^k)$

not feasible time for many long sequences…

58

## Multiple Alignment: Running Time

- For 3 sequences of length $n$, the run time is proportional to $7n^3$ *(each of n cubed node has 7 neighbors to choose from).*
- For a $k$-way alignment, build a $k$-dimensional Manhattan graph with
  - $n^k$ nodes
  - most nodes have $2^k - 1$ incoming edges.
  - Runtime: $O(2^k n^k)$

Biologists use fast heuristic algorithms (don't guarantee the final optimal solutions, but they guarantee that this algorithm will finish at a meaningful time!)

59

## Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments:

```
A C - G C G G - C
A C - G C - G A G
G C C G C - G A G
```



```
ACGCGG-C    AC-GCGG-C    AC-GCGAG
ACGC-GAC    GCCGC-GAG    GCCGCGAG
```

60

## Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments:

```
A C - G C G G - C
A C - G C - G A G
G C C G C - G A G
```

```
ACGCGG-C    AC-GCGG-C    AC-GCGAG
ACGC-GAC    GCCGC-GAG    GCCGCGAG
```

**STOP:** Vice versa? In general: no…

61

## Idea: Construct Multiple from Pairwise Alignments?

Given a set of **arbitrary** pairwise alignments, can we construct a multiple alignment that induces them?

Given `AAAATTTT`, `TTTTGGGG`, `GGGGAAAA`,
we can consider these 3 pairwise alignments (4 matches)

```
AAAATTTT----      ----AAAATTTT      TTTTGGGG----
----TTTTGGGG      GGGGAAAA----      ----GGGGAAAA
```

62

## Idea: Construct Multiple from Pairwise Alignments?

Given a set of **arbitrary** pairwise alignments, can we construct a multiple alignment that induces them?

Given `AAAATTTT`, `TTTTGGGG`, `GGGGAAAA`,

we can consider these 3 pairwise alignments (4 matches)

```
AAAATTTT----      ----AAAATTTT      TTTTGGGG----
----TTTTGGGG      GGGGAAAA----      ----GGGGAAAA
```

…but we cannot construct a multiple alignment that induces these 3 pairwise alignments

63

## Greedy Heuristic for Multiple Alignment ☹

1. Find an optimal pairwise alignment of each pair of strings.
2. Combine the set of optimal pairwise alignments into a multiple alignment.

There is no way to combine these optimal pairwise alignment into a meaningful multiple alignment!

```
AAAATTTT----      ----AAAATTTT      TTTTGGGG----
----TTTTGGGG      GGGGAAAA----      ----GGGGAAAA
```

64

## Profile Representation of Multiple Alignment

65

## Profile Representation of Multiple Alignment

A *profile* is a probability for each letter to occur in each column, i.e., a description of the consensus of a multiple sequence alignment. It uses a position-specific scoring system to capture information about the degree of conservation at various positions in the multiple alignment.

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

A  0  1  0  0  0  0  1  0  0 .8  0  0  0  0
C .6  0  0  0  1  0  0 .4  1  0 .6 .2  0  0
G  0  0  1 .2  0  0  0  0  0 .2  0  0 .4  1
T .2  0  0  0  0  1  0 .6  0  0  0  0 .2  0
- .2  0  0 .8  0  0  0  0  0  0 .4 .8 .4  0
```

p=1/#

66

## Aligning Sequence Against Sequence

- In the past we were aligning a **sequence against a sequence**.

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

A         0  1  0  0  0  0  1  0  0 .8  0  0  0  0
C        .6  0  0  0  1  0  0 .4  1  0 .6 .2  0  0
G         0  0  1 .2  0  0  0  0  0 .2  0  0 .4  1
T        .2  0  0  0  0  1  0 .6  0  0  0  0 .2  0
-        .2  0  0 .8  0  0  0  0  0  0 .4 .8 .4  0
```

67

## Aligning Sequence Against Profile

- In the past we were aligning a **sequence against a sequence**.
  - Can we align a **sequence against a profile?**

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

A         0  1  0  0  0  0  1  0  0 .8  0  0  0  0
C        .6  0  0  0  1  0  0 .4  1  0 .6 .2  0  0
G         0  0  1 .2  0  0  0  0  0 .2  0  0 .4  1
T        .2  0  0  0  0  1  0 .6  0  0  0  0 .2  0
-        .2  0  0 .8  0  0  0  0  0  0 .4 .8 .4  0
```

68

## Aligning Profile Against Profile

- In the past we were aligning a **sequence against a sequence.**
  - Can we align a **sequence against a profile?**
  - Can we align a **profile against a profile?**

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

A   0  1  0  0  0  0  1  0  0 .8  0  0  0  0
C  .6  0  0  0  1  0  0 .4  1  0 .6 .2  0  0
G   0  0  1 .2  0  0  0  0  0 .2  0  0 .4  1
T  .2  0  0  0  0  1  0 .6  0  0  0  0 .2  0
-  .2  0  0 .8  0  0  0  0  0  0 .4 .8 .4  0
```

69

## Greedy Progressive Multiple Alignment

- Choose the most *similar* sequences and combine them into a profile, thereby reducing alignment of *k* sequences to an alignment of (*k* – 2) sequences and 1 profile.
- Iterate

70

## Greedy Approach: Example

- Sequences: GATTCA, GTCTGA, GATATT, GTCAGC.
- 6 pairwise alignments (premium for match +1, penalties for indels and mismatches -1)

```
s2  GTCTGA                 s1  GATTCA--
s4  GTCAGC  (score = 2)    s4  G-T-CAGC (score = 0)

s1  GAT-TCA                s2  G-TCTGA
s2  G-TCTGA (score = 1)    s3  GATAT-T  (score = -1)

s1  GAT-TCA                s3  GAT-ATT
s3  GATAT-T (score = 1)    s4  G-TCAGC  (score = -1)
```

71

## Greedy Approach: Example

- Since $s_2$ and $s_4$ are closest, we consolidate them into a profile:

$$\left.\begin{matrix} s2 & \text{GTCTGA} \\ s4 & \text{GTCAGC} \end{matrix}\right\} \; s_{2,4} = \text{GTCt/aGa/c}$$

- New set of 3 sequences to align:

| | |
|---|---|
| $s_1$ | GATTCA |
| $s_3$ | GATATT |
| $s_{2,4}$ | GTCt/aGa/c |

- … and so on. This algorithm may be not optimal but it is fast.

72

## Are these concepts similar?

- Profiles / pan-genome graphs
- Profiles / degenerate strings (Regular expression)

73

## HIV AND ALIGNING A NEW GENE AGAINST A KNOWN FAMILY

74

## Waiting for an HIV Vaccine …

Yet another terrible disease is about to yield to patience, persistence and outright genius.

**Margaret Heckler**
**1984**

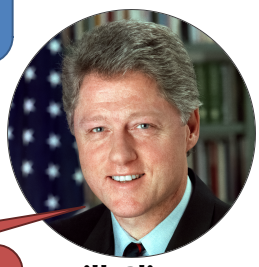
75

## Waiting for an HIV Vaccine …

Yet another terrible disease is about to yield to patience, persistence and outright genius.

**Margaret Heckler**
**1984**

It is no longer a question of *whether* we can develop an AIDS vaccine, it is simply a question of *when*.



**Bill Clinton**
**1997**

76

# Waiting for an HIV Vaccine …

The failed HIV Merck vaccine study: a step back or a launching point for future vaccine development?

Rafick-Pierre Sekaly

‣ Author information ‣ Copyright and License information Disclaimer

See the article "An HIV-1 clade C DNA prime, NYVAC boost vaccine regimen induces reliable, polyfunctional, and long-lasting T cell responses" on page 63.

This article has been cited by other articles in PMC.

## Abstract

Go to:

The world of human immunodeficiency virus (HIV) vaccines has suffered a baffling setback. The first trial of a vaccine designed to elicit strong cellular immunity has shown no protection against infection. More alarmingly, the vaccine appeared to increase the rate of HIV infection in individuals with prior immunity against the adenovirus vector used in the vaccine. A new study in this issue suggests that a different vaccine approach—using a DNA prime/poxvirus boost strategy—induces polyfunctional immune responses to an HIV immunogen. The disappointing results of the recent vaccine trial suggest that a more thorough assessment of vaccine-induced immune responses is urgently needed, and that more emphasis should be placed on primate models before efficacy trials are undertaken.

77

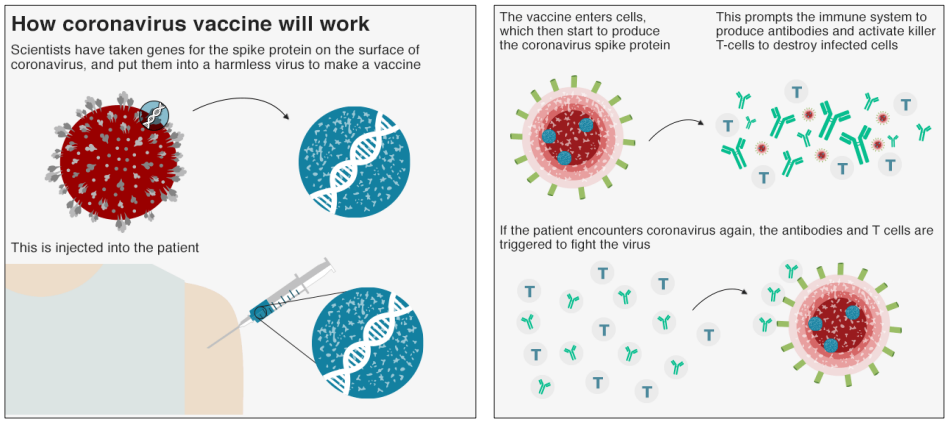
# … and yet we have a coronavirus vaccine in under a year???


Citation: O.J. "The Juice" Simpson Twitter account

78

# Many Vaccines Target Viral Surface Proteins

Astrazeneca

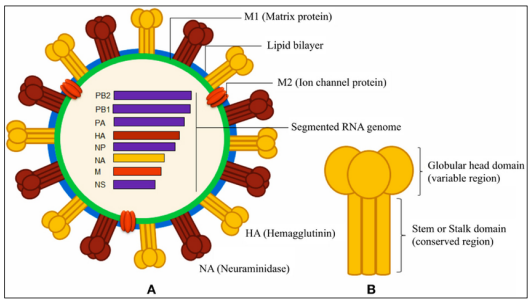



Citation: https://www.bbc.com/news/health-52394485

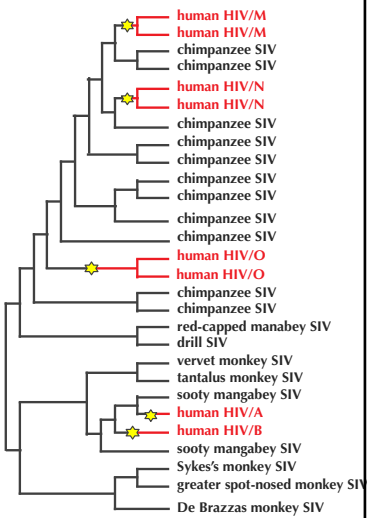
79

# Many Vaccines Target Viral Surface Proteins

Vaccines training the immune system to recognize HIV's surface proteins fail because HIV strains are so *variable*.




https://www.frontiersin.org/articles/10.3389/fimmu.2015.00336/full

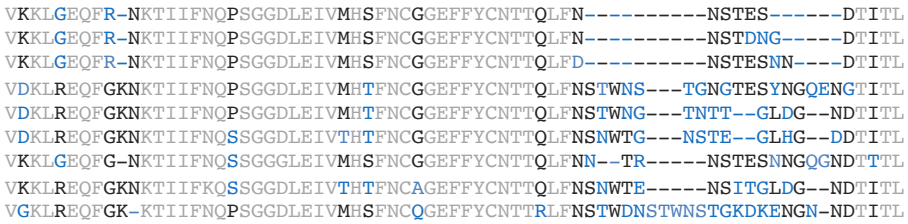



80

## HIV Drug "Cocktails" Have to Deal with Variability

The HIV population in a ***single*** infected individual rapidly evolves to evade the immune system.

envelope glycoprotein gp120

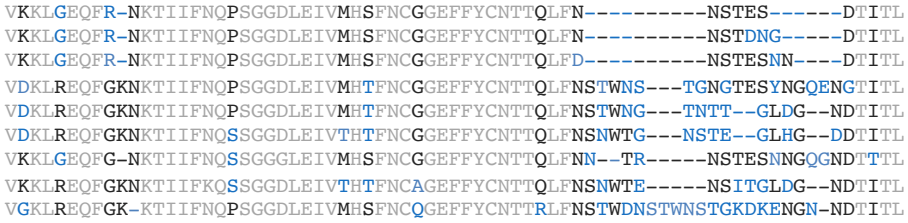



81

## HIV Drug "Cocktails" Have to Deal with Variability

The HIV population in a ***single*** infected individual rapidly evolves to evade the immune system.

envelope glycoprotein gp120




HIV strains from ***different*** patients are diverged phenotypes requiring different drug cocktails.

82

## Returning to Multiple Alignment

**Multiple Alignment Problem:** *Find the highest-scoring alignment between multiple strings.*

- **Input:** A collection of *t* strings (and some way of scoring columns of a multiple alignment).
- **Output:** A multiple alignment of these strings having maximum score.

A single misalignment could lead to an error, so we have to be accurate. And so we need a *problem formulation* that scores different columns differently (we align a new sequence to the hystory of the virus).

83

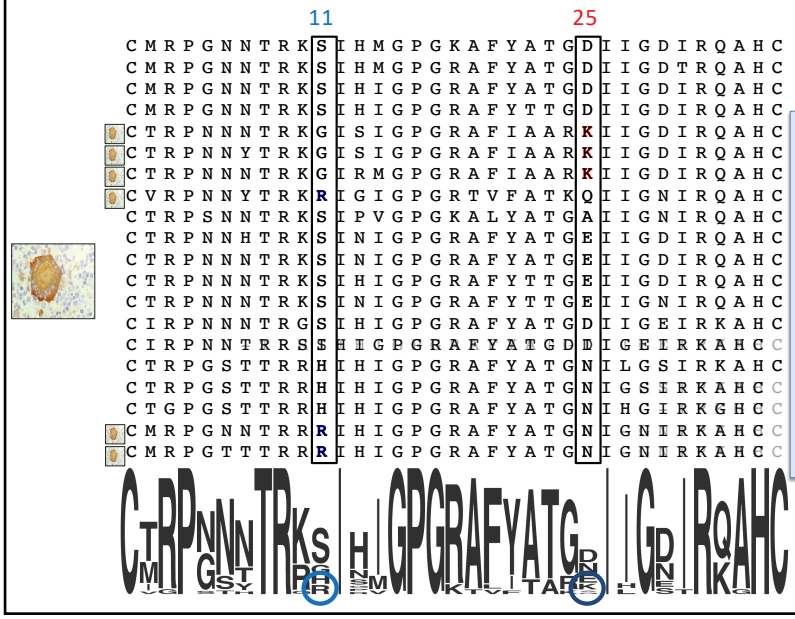
## Motivation

Once we have a collection of *known* protein alignments ("families"), we need to be able to identify which family a new protein belongs to. That is, add **a** new string into **an existing alignment**.

```
VKKLGEQFR-NKTIIFNQPSGGDLEIVMHSFNCGGEFFYCNTTQLFN----------NSTES------DTITL
VKKLGEQFR-NKTIIFNQPSGGDLEIVMHSFNCGGEFFYCNTTQLFN----------NSTDNG-----DTITL
VKKLGEQFR-NKTIIFNQPSGGDLEIVMHSFNCGGEFFYCNTTQLFD----------NSTESNN----DTITL
VDKLREQFGKNKTIIFNQPSGGDLEIVMHTFNCGGEFFYCNTTQLFNSTWNS---TGNGTESYNGQENGTITL
VDKLREQFGKNKTIIFNQPSGGDLEIVMHTFNCGGEFFYCNTTQLFNSTWNG---TNTT--GLDG--NDTITL
VDKLREQFGKNKTIIFNQSSGGDLEIVTHTFNCGGEFFYCNTTQLFNSNWTG---NSTE--GLHG--DDTITL
VKKLGEQFG-NKTIIFNQSSGGGLEIVMHSFNCGGEFFYCNTTQLFNN--TR-----NSTESNNGQGNDTTTL
VKKLREQFGKNKTIIFKQSSGGDLEIVTHTFNCAGEFFYCNTTQLFNSNWTE-----NSITGLDG--NDTITL
VGKLREQFGK-KTIIFNQPSGGDLEIVMHSFNCQGEFFYCNTTRLFNSTWDNSTWNSTGKDKENGN-NDTITL
```

84

## Identifying Syncytium-Inducing HIVs: **11-25 Rule**




**11-25 rule**: SI HIVs have **R** or **K** at positions 11 or 25 in the V3 region (envelope protein)

85

## A Limitation of Sequence Alignment

To predict an HIV phenotype, we need an *accurate* alignment: a single misalignment at a position influencing the SI phenotype leads to an error (e.g. **11-25 Rule)**.

**STOP and Think:** Was it a good idea to use the *same* scoring matrix across different columns of an alignment?

We need a statistically solid **problem formulation** for alignment that uses a *different* scoring approach at different columns.

86

## Is the "Profile representation" mentioned above suitable?

To predict an HIV phenotype, we need an *accurate* alignment: a single misalignment at a position influencing the SI phenotype leads to an error (e.g. **11-25 Rule)**.

**STOP and Think:** Was it a good idea to use the *same* scoring matrix across different columns of an alignment?

We need a statistically solid **problem formulation** for alignment that uses a *different* scoring approach at different columns.

87

## Hidden Markov Model (HMM)

88

# Hidden Markov Model (HMM)

Σ: an **alphabet** of emitted symbols — H and T

*States* : a set of **hidden states** — *F* and *B*

*Transition* = ($transition_{l,k}$): a |*States*| ×|*States*| matrix of **transition probabilities** (of changing from state *l* to state *k*)

| | *F* | *B* |
|---|---|---|
| *F* | 0.9 | 0.1 |
| *B* | 0.1 | 0.9 |

*Emission*= ($emission_k(b)$): a |*States*| × |∑| matrix of **emission probabilities** (of emitting symbol *b* when the HMM is in state *k*)

| | H | T |
|---|---|---|
| *F* | 0.50 | 0.50 |
| *B* | 0.75 | 0.25 |

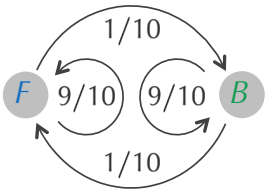
Goal: Infer the most likely sequence of hidden states based on the sequence of emitted symbols

89

# HMM Diagram

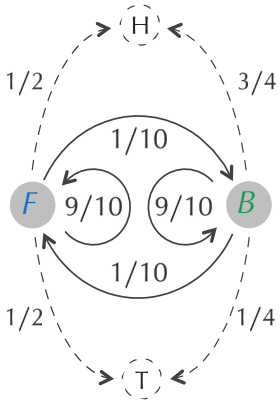
*Transition*

| | *F* | *B* |
|---|---|---|
| *F* | 0.9 | 0.1 |
| *B* | 0.1 | 0.9 |




90

# HMM Diagram




*Transition*

| | *F* | *B* |
|---|---|---|
| *F* | 0.9 | 0.1 |
| *B* | 0.1 | 0.9 |

*Emission*

| | H | T |
|---|---|---|
| *F* | 0.50 | 0.50 |
| *B* | 0.75 | 0.25 |

91

# Hidden Path

**Hidden path:** the sequence $\pi = \pi_1 \ldots \pi_n$ of states that the HMM passes through.

- Pr($x$, $\pi$): the probability that an HMM follows the hidden path $\pi$ *and* emits the string $x = x_1 x_2 \ldots x_n$.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x$: | T | H | T | H | H | H | T | H | T | T | H |
| $\pi$: | F | F | F | B | B | B | B | B | F | F | F |

$$\sum_{\text{all possible emitted strings } x} \sum_{\text{all possible hidden paths } \pi} \Pr(x, \pi) = 1$$

- Pr($x|\pi$): the **conditional probability** that an HMM emits the string $x$ *after* following the hidden path $\pi$.

$$\sum_{\text{all possible emitted strings } x} \Pr(x|\pi) = 1$$

92

$$\Pr(x, \pi) = \Pr(x|\pi) * \Pr(\pi)$$

- $\Pr(x, \pi)$: the probability that an HMM follows the hidden path $\pi$ and emits the string $x$.
- $\Pr(x_i|\pi_i)$ – probability that $x_i$ was emitted from the state $\pi_i$ (equal to $emission_{\pi i}(x_i)$).
- $\Pr(\pi_{i-1} \rightarrow \pi_i)$ – probability that the HMM moved from $\pi_{i-1} \rightarrow \pi_i$ (equal to $transition_{\pi i, \pi i+1}$).

| $x$ | | T | H | T | H | H | H | T | H | T | T | H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\pi$ | | F | F | F | B | B | B | B | B | F | F | F |
| $\Pr(\pi_{i-1} \rightarrow \pi_i)$ | .5 | .9 | .9 | .1 | .9 | .9 | .9 | .9 | .1 | .9 | .9 | |
| $\Pr(x_i\|\pi_i)$ | ½ | ½ | ½ | ¾ | ¾ | ¾ | ¼ | ¾ | ½ | ½ | ½ | |

$$\Pr(\pi) = \Pi_{i=1,n} \Pr(\pi_{i-1} \rightarrow \pi_i) = \Pi_{i=1,n} transition_{\pi i-1, \pi i}$$

$$\Pr(x|\pi) = \Pi_{i=1,n} \Pr(x_i|\pi_i) = \Pi_{i=1,n} emission_{\pi i}(x_i)$$

93

## Decoding Problem… Viterbi algorithm

**Decoding Problem**: *Find an optimal hidden path in an HMM given its emitted string.*

- **Input:** A string $x = x_1 \ldots x_n$ emitted by an HMM (∑, *States, Transition, Emission*).
- **Output:** A path $\pi$ that maximizes the probability $\Pr(x,\pi)$ over all possible paths through this HMM.

$$\begin{aligned} \Pr(x, \pi) &= \Pr(x|\pi) * \Pr(\pi) \\ &= \Pi_{i=1,n} \Pr(x_i|\pi_i) * \Pr(\pi_{i-1} \rightarrow \pi_i) \\ &= \Pi_{i=1,n} emission_{\pi i}(x_i) * transition_{\pi i-1, \pi i} \end{aligned}$$

94