An introduction on bioinformatics

Quentin NUGIER
MICALIS institute, DynPhage team

Bioinformatics

Biology -> a lot of datas, analysis, questions to answer
-> a lot of possible tasks

Bioinformatics

Biology -> a lot of datas, analysis, questions to answer
-> a lot of possible tasks

For each task, multiple tools available
Each tool has its specificity (approach, organism…)

Bioinformatics

Biology -> a lot of datas, analysis, questions to answer
-> a lot of possible tasks

For each task, multiple tools available
Each tool has its specificity (approach, organism…)

And each tool is available in multiple working environnements :

- multiple programming languages (python, R, bash…)
- multiple services (galaxy, google colab…)

Bioinformatics

Biology -> a lot of datas, analysis, questions to answer
-> a lot of possible tasks

For each task, multiple tools available
Each tool has its specificity (approach, organism…)

And each tool is available in multiple working environnements :

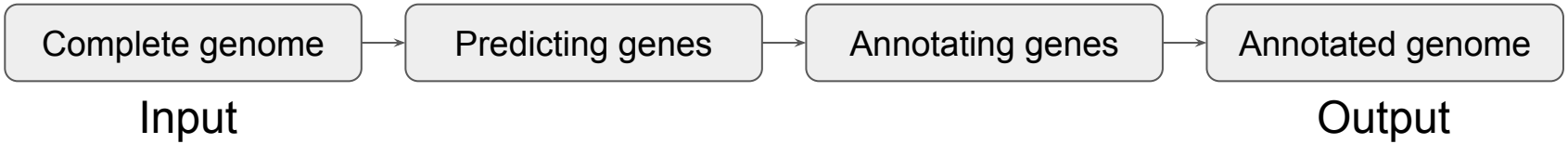
- multiple programming languages (python, R, bash…)
- multiple services (galaxy, google colab…)

=> For today, we'll greatly limit our overview to a specific workflow

For today

Today’s goal: knowing how to annotate a phage genome

Workflow :
execution of multiple computational/data manipulation steps



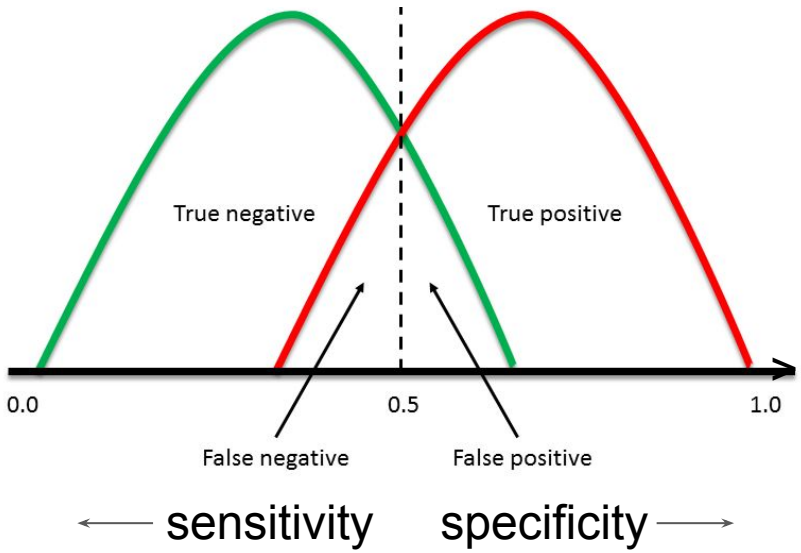
Detecting proteins : gene calling

We want to detect genes (ORFs) -> protein coding DNA sequences (CDS)
A process called “gene calling” :
Detect all true ORFs (true positives) = sensitivity
NOT detect all wrong ORFs (true negatives) = specificity

Detecting proteins : gene calling

We want to detect genes (ORFs) -> protein coding DNA sequences (CDS)
A process called “gene calling” :
Detect all true ORFs (true positives) = sensitivity
NOT detect all wrong ORFs (true negatives) = specificity

Trade-off :

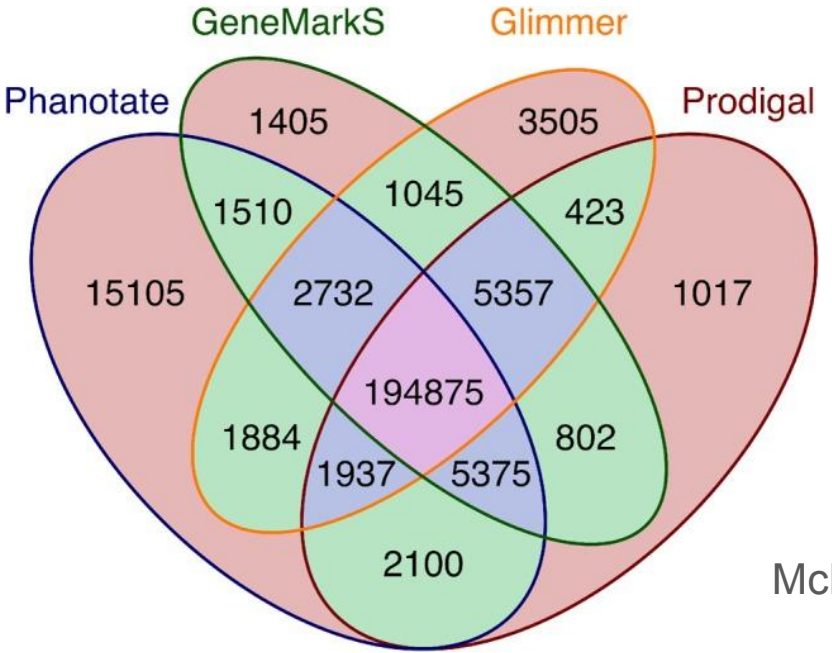


vertical line = threshold of detection

stringency

Detecting proteins : gene calling

Multiple available tools : Prodigal, Phanotate, GeneMarkS, Glimmer…
Each giving different results :



McNair et al. 2019

Detecting proteins : gene calling

Why different results ?
Multiple metrics to predict Open Reading Frames (**ORF)** :
GC% in codons, starting codons detection (including alternative genetic code), overlapping constraints, length constraints, ribosome binding site

Detecting proteins : gene calling

Why different results ?
Multiple metrics to predict Open Reading Frames (**ORF)** :
GC% in codons, starting codons detection (including alternative genetic code), overlapping constraints, length constraints, ribosome binding site

A comparison of prodigal and phanotate :

	Prodigal	Phanotate
Organism	Prokaryotes	Phages

Detecting proteins : gene calling

Why different results ?
Multiple metrics to predict Open Reading Frames (**ORF)** :
GC% in codons, starting codons detection (including alternative genetic code), overlapping constraints, length constraints, ribosome binding site

A comparison of prodigal and phanotate :

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORFs	Less	More

Detecting proteins : gene calling

Why different results ?
Multiple metrics to predict Open Reading Frames (**ORF)** :
GC% in codons, starting codons detection (including alternative genetic code), overlapping constraints, length constraints, ribosome binding site

A comparison of prodigal and phanotate :

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORFs	Less	More
ORF size on average	Longer	Shorter

Detecting proteins : gene calling

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORF	Less	More
ORF size on average	Longer	Shorter
Specificity		
Sensitivity		

Detecting proteins : gene calling

From prodigal : “In the construction of a novel algorithm, we determined it would be preferable to sacrifice some genuine predictions if it meant also eliminating a much larger number of false identifications.” Hyatt et al. 2010

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORF	Less	More
ORF size on average	Longer	Shorter
Specificity	Higher	
Sensitivity	Lower	

Detecting proteins : gene calling

From prodigal : “In the construction of a novel algorithm, we determined it would be preferable to sacrifice some genuine predictions if it meant also eliminating a much larger number of false identifications.” Hyatt et al. 2010
From phanotate “We designed a completely novel approach [...] to minimize non-coding DNA bases” McNair et al. 2019

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORF	Less	More
ORF size on average	Longer	Shorter
Specificity	Higher	Lower
Sensitivity	Lower	Higher

Detecting proteins : gene calling

Practical choice: Prefer Prodigal for cleaner annotation

	Prodigal	Phanotate
Organism	Prokaryotes	Phages
Number of ORF	Less	More
ORF size on average	Longer	Shorter
Specificity	Higher	Lower
Sensitivity	Lower	Higher

Annotating protein function : the notion of homologs

Now that we have the ORFs
How to predict function of the proteins ?

Annotating protein function : the notion of homologs

Now that we have the ORFs
How to predict function of the proteins ?

A bit of linguistics :

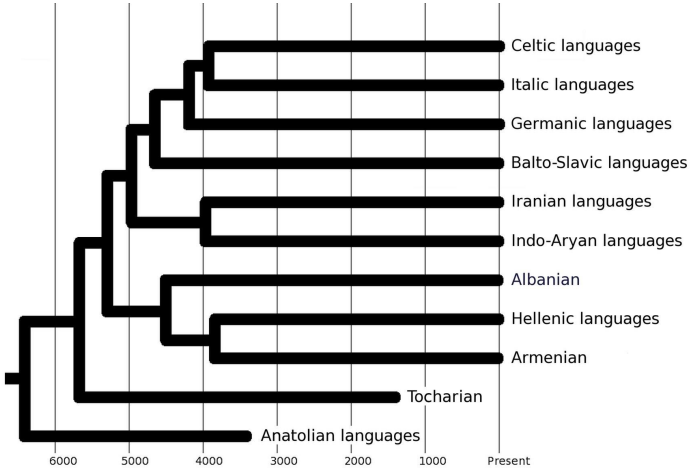
In English this fruit is an apricot 
In French : abricot
German : aprikose
Italian : albicocca

Annotating protein function : the notion of homologs

Now that we have the ORFs
How to predict function of the proteins ?

A bit of linguistics :

In English this fruit is an apricot
In French : abricot
German : aprikose
Italian : albicocca



The closer the words, the closer the languages (which evolved from common ancestry).

Annotating protein function : the notion of homologs

Proteins having a common ancestor are homologous -> related sequences have related functions :
Find the closest related annotated sequences to our ORF, and transfer the annotation.

Annotating protein function : the notion of homologs

Proteins having a common ancestor are homologous -> related sequences have related functions :
Find the closest related annotated sequences to our ORF, and transfer the annotation.

If we want to find the closest related word to “apricot” among

abricot
aprikose
albicocca

Annotating protein function : the notion of homologs

Proteins having a common ancestor are homologous -> related sequences have related functions :
Find the closest related annotated sequences to our ORF, and transfer the annotation.

If we want to find the closest related word to “apricot” among

abricot
aprikose
albicocca

When comparing sequences, identical amino acids must be rewarded, whereas different amino acids and gaps should be penalized.

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot

abricot
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|
abricot
+1              =
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|.
abricot
+1-1          =
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|.|
abricot
+1-1+1          =
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|.|||||
abricot
+1-1+1+1+1+1+1=5
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|.|||||
abricot
+1-1+1+1+1+1+1=5
```

```
apricot-
||||.|.-
aprikose
+1+1+1+1-1+1-1-1=2
```

Comparing sequences

Scoring system for alignment:
+1 for identical amino acid
-1 for different amino acid
-1 for gap

```
apricot
|.|||||
abricot
+1-1+1+1+1+1+1=5
```

```
apricot-
||||.|.-
aprikose
+1+1+1+1-1+1-1-1=2
```

By scoring, we can detect the closest word = the closest protein.

Comparing sequences

Before scoring, we need the best alignment

How to align apricot to albaricocca ?

Comparing sequences

Before scoring, we need the best alignment

How to align apricot to albaricocca ?

There is a lot possible alignments, from the most correct to the least.

```
---apricot--    -apricot-
---|-||||.--    -.......---
alba-ricocca    albaricocca

score=-1         score=-7
```

Comparing sequences

Before scoring, we need the best alignment

How to align apricot to albaricocca ?

There is a lot possible alignments, from the most correct to the least.

```
---apricot--    -apricot-
---|-||||.--    -.......---
alba-ricocca    albaricocca

score=-1         score=-7
```

Finding the best alignment is also a problem

Alignments

Try all alignments -> too long

-> heuristic method

↳ Speeding up a process of finding a satisfactory solution

Alignments

Try all alignments -> too long

-> heuristic method

↳ Speeding up a process of finding a satisfactory solution

Needleman-Wunsch algorithm: find the best score by using only the best solution locally.

This algorithm is used to align every amino acids : global alignment.

Alignments

Try all alignments -> too long

-> heuristic method

↳ Speeding up a process of finding a satisfactory solution

Needleman-Wunsch algorithm: find the best score by using only the best solution locally.

This algorithm is used to align every amino acids : global alignment.

BUT

```
apricot-
||||.|.-
aprikose
```

is not the best alignment

Alignments

The last two characters are not aligned properly :

```
apricot-
||||.|.-    score=2
aprikose
```

Alignments

The last two characters are not aligned properly :

```
apricot-
||||.|.-    score=2
aprikose
```

Therefore, a better alignment would be one that ignores them :

```
apricot
||||.|      score=4
aprikose
```

Alignments

The last two characters are not aligned properly :

```
apricot-
||||.|.-    score=2
aprikose
```

Therefore, a better alignment would be one that ignores them :

```
apricot
||||.|      score=4
aprikose
```

Aligns best matching regions : local alignment

Alignments

```
apricot          apricot
||||.|           |-||||
aprikose      alba-ricocca
score=4          score=4
```

Smith–Waterman algorithm

Alignments

```
apricot          apricot
||||.|           |-||||
aprikose     alba-ricocca
score=4          score=4
```

Smith–Waterman algorithm

Local alignments are what’s used for protein comparison.

Two more metrics to describe an alignment :
-> Identity percentage
-> Coverage percentage

Alignment : identity percentage

The identity percentage is the percentage of identical amino acids among those that are aligned :

Alignment : identity percentage

The identity percentage is the percentage of identical amino acids among those that are aligned :

```
apricot
||||.|
aprikose

score=4

5/6=83% identity
```

Alignment : coverage

The coverage percentage is the number of amino acids aligned divided by the total length of the sequence (including what's not aligned):

```
apricot          coverage of apricot: 6/7=86%
||||.|
aprikose         coverage of aprikose: 6/8=75%

score=4
```

Alignment : coverage

The coverage percentage is the number of amino acids aligned divided by the total length of the sequence (including what's not aligned):

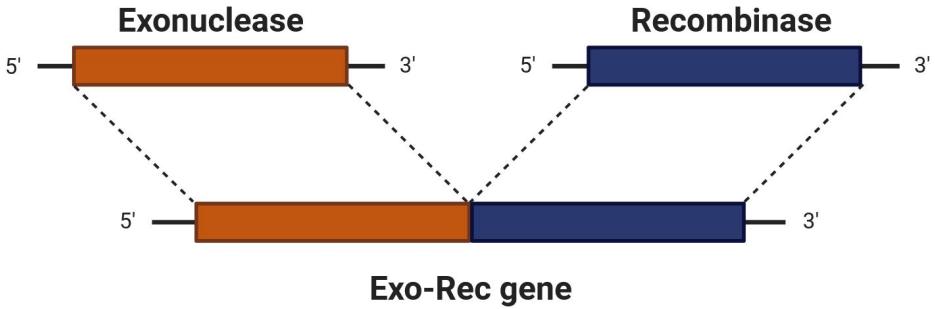
```
apricot          coverage of apricot: 6/7=86%
||||.|
aprikose         coverage of aprikose: 6/8=75%

score=4
```

Two different coverage percentages.
Which one should you use ?

Alignment : coverage

In a lot of cases, both !
A **reciprocal coverage** is frequently used

Why ? An example with a protein fusion:

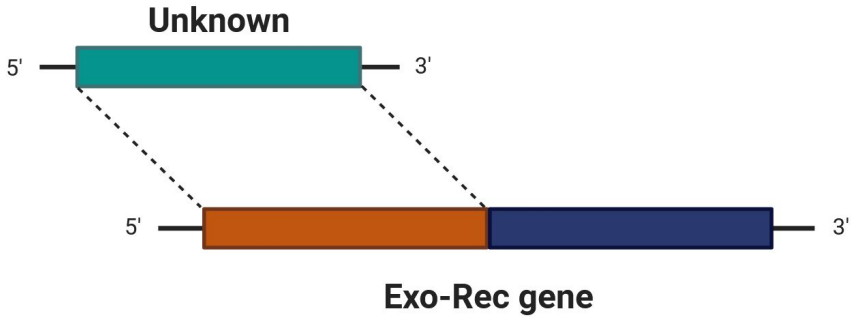


Alignment : coverage

In a lot of cases, both !
A **reciprocal coverage** is frequently used

Why ? An example with a protein fusion:

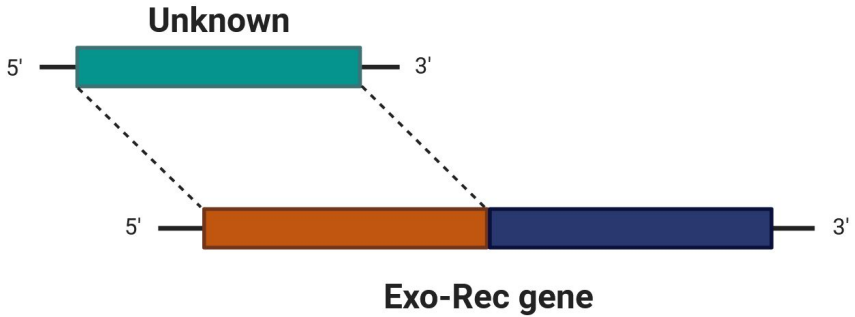
Coverage on unknown : 100 %
Coverage on Exo-Rec : 50 %



Alignment : coverage

In a lot of cases, both !
A **reciprocal coverage** is frequently used

Why ? An example with a protein fusion:



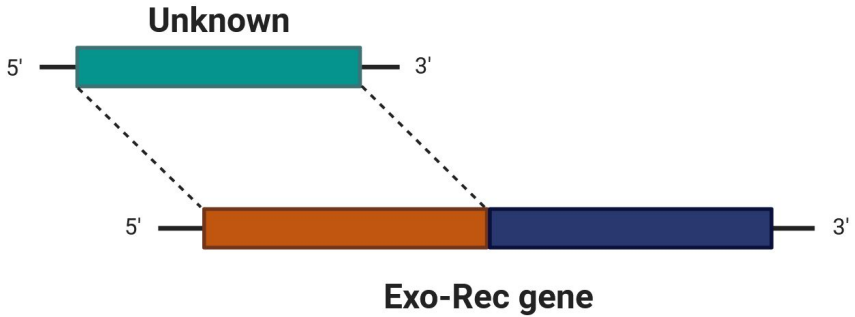
Coverage on unknown : 100 %
Coverage on Exo-Rec : 50 %

No threshold (0%) = Homology inferred -> Exo-Rec -> wrong annotation ! ❌

Alignment : coverage

In a lot of cases, both !
A **reciprocal coverage** is frequently used

Why ? An example with a protein fusion:



Coverage on unknown : 100 %
Coverage on Exo-Rec : 50 %

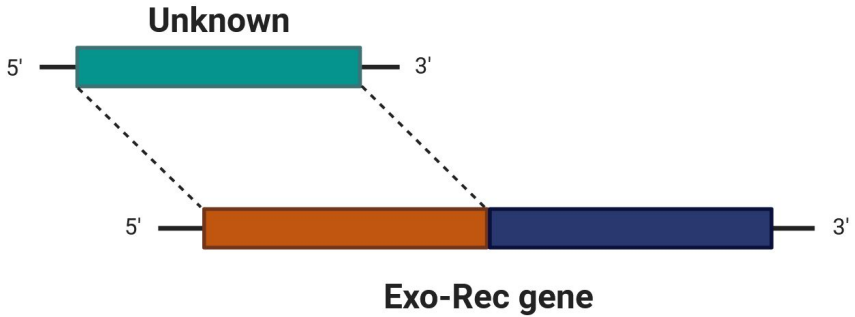
No threshold (0%) = Homology inferred -> Exo-Rec -> wrong annotation !

70 % threshold = Homology not inferred -> no wrong annotation

Alignment : coverage

In a lot of cases, both !
A **reciprocal coverage** is frequently used

Why ? An example with a protein fusion:



Coverage on unknown : 100 %
Coverage on Exo-Rec : 50 %

No threshold (0%) = Homology inferred -> Exo-Rec -> wrong annotation !

70 % threshold = Homology not inferred -> no wrong annotation

What do we compare them to -> NCBI with BLAST

BLAST

“basic local alignment search tool”

Rank: **12** Citations: **38,380**

Basic local alignment search tool.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J.

J. Mol. Biol. **215**, 403–410 (1990).

BLAST

“basic local alignment search tool”

Rank: **12** Citations: **38,380**

Basic local alignment search tool.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J.

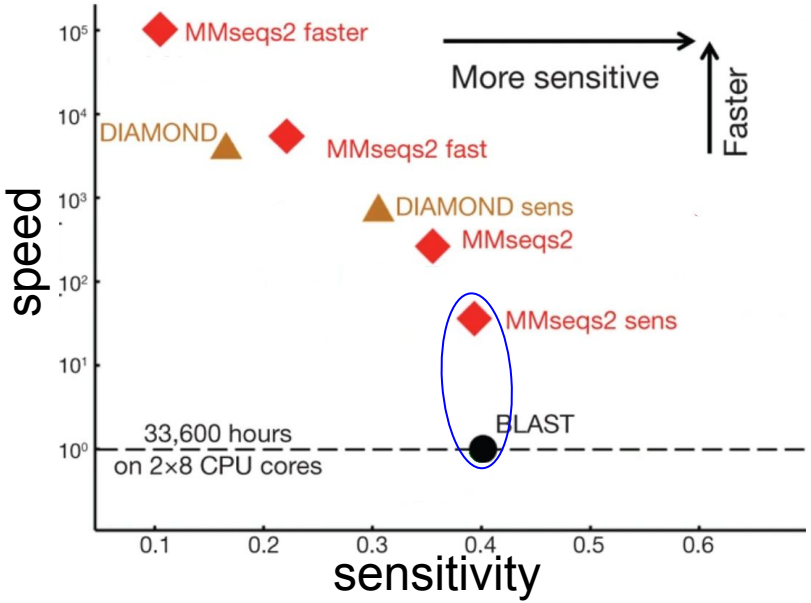
J. Mol. Biol. **215**, 403–410 (1990).

Allows for the search of a **query** sequence against a **database (NCBI)** of hundreds of thousands sequences in just a few seconds/minutes. The database is a collection of **target (=subject)** sequences**.**

Nature	Program	Query	Database
Nucleotide BLAST	blastn	Nucleotide (DNA, RNA)	Nucleotide (DNA, RNA)
Protein BLAST	blastp	Protein	Protein

BLAST

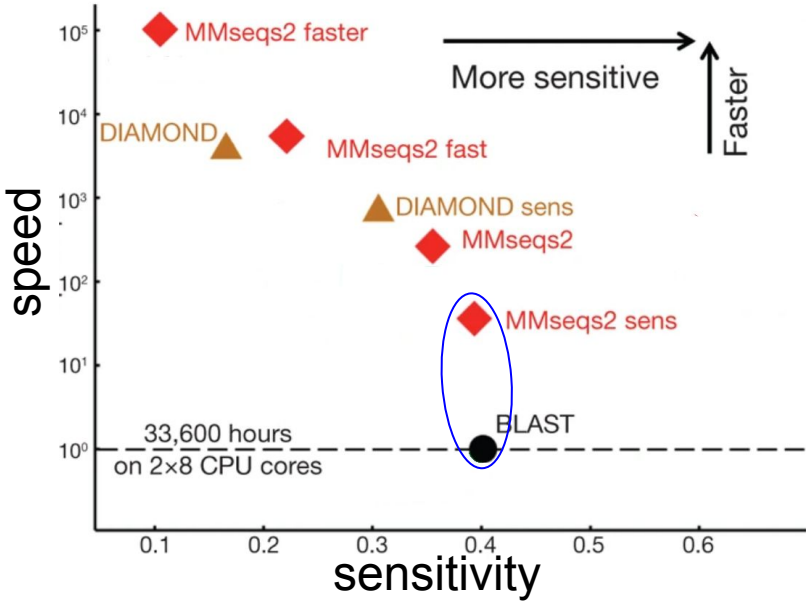
BLAST is the most popular sequence comparison/alignment tool
But others are available : DIAMOND and MMseqs2



Adapted from Steinegger and Söding, 2017

BLAST

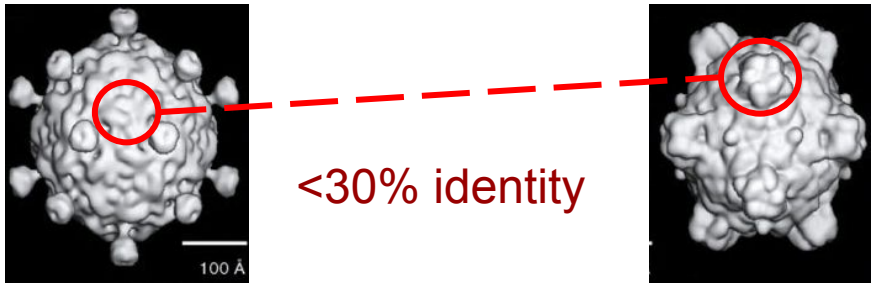
BLAST is the most popular sequence comparison/alignment tool
But others are available : DIAMOND and MMseqs2



Adapted from Steinegger and Söding, 2017

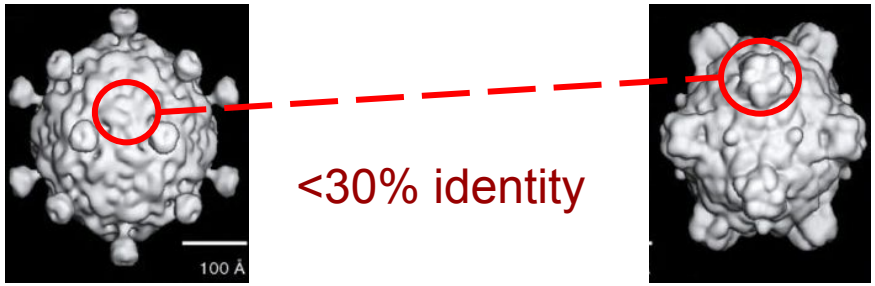
Trade-off between speed and sensitivity, ubiquitous in bioinformatics

BLAST



Two microviridae : SpV4 PhiX174

BLAST



Dokland et al. 1997

Two microviridae : SpV4 PhiX174

Sequence comparison tools can’t detect homologous proteins that are too divergent

-> Need of a more sensitive method

Further than blast : sequence-profile comparison

Pairwise alignments (one against one) -> Multiple Sequence Alignment (MSA)

Further than blast : sequence-profile comparison

Pairwise alignments (one against one) -> Multiple Sequence Alignment (MSA)

MSA of apricot, abricot, aprikose:

```
apricot-
abricot-
aprikose
```

Further than blast : sequence-profile comparison

Pairwise alignments (one against one) -> Multiple Sequence Alignment (MSA)

MSA of apricot, abricot, aprikose:

```
apricot-
abricot-
aprikose
```

The information of all sequences is inside the MSA
Build the profile of these MSAs to refine the information contained in apricot

Further than blast : sequence-profile comaprison

First simple model : PSSM (position specific score matrix)
A simple **matrix (=table)** with positions as columns and amino acid in lines

```
apricot-
abricot-
aprikose
```

Further than blast : sequence-profile comparison

First simple model : PSSM (position specific score matrix)
A simple **matrix (=table)** with positions as columns and amino acid in lines

apricot-
abricot-
aprikose

→

	1	2	3	4	5	6	7	8
a	+10	0	0	0	0	0	0	0
b	0	+3	0	0	0	0	0	0
p	0	+4	0	0	0	0	0	0
r	0	0	+10	0	0	0	0	0
i	0	0	0	+10	0	0	0	0
c	0	0	0	0	+4	0	0	0
k	0	0	0	0	+3	0	0	0
o	0	0	0	0	0	+10	0	0
t	0	0	0	0	0	0	+4	0
s	0	0	0	0	0	0	+3	0
e	0	0	0	0	0	0	0	+10

Further than blast : sequence-profile comparison

First simple model : PSSM (position specific score matrix)
A simple **matrix (=table)** with positions as columns and amino acid in lines

```
apricot-
abricot-
aprikose
```

	1	2	3	4	5	6	7	8
a	+10	0	0	0	0	0	0	0
b	0	+3	0	0	0	0	0	0
p	0	+4	0	0	0	0	0	0
r	0	0	+10	0	0	0	0	0
i	0	0	0	+10	0	0	0	0
c	0	0	0	0	+4	0	0	0
k	0	0	0	0	+3	0	0	0
o	0	0	0	0	0	+10	0	0
t	0	0	0	0	0	0	+4	0
s	0	0	0	0	0	0	+3	0
e	0	0	0	0	0	0	0	+10

Further than blast : sequence-profile comparison

Compare a sequence to a PSSM (PSI-BLAST)
“sequence-profile” comparison

apricot vs

Further than blast : sequence-profile comparison

Compare a sequence to a PSSM (PSI-BLAST)
“sequence-profile” comparison

apricot vs 

At the position 2, there is both a “b” and a “p”

Further than blast : sequence-profile comparison

Compare a sequence to a PSSM (PSI-BLAST)
“sequence-profile” comparison

apricot **vs** 

At the position 2, there is both a “b” and a “p”

-> more information than in a single sequence
-> a sequence having either a “b” or a “p” would be a match

Further than blast : sequence-profile comparison

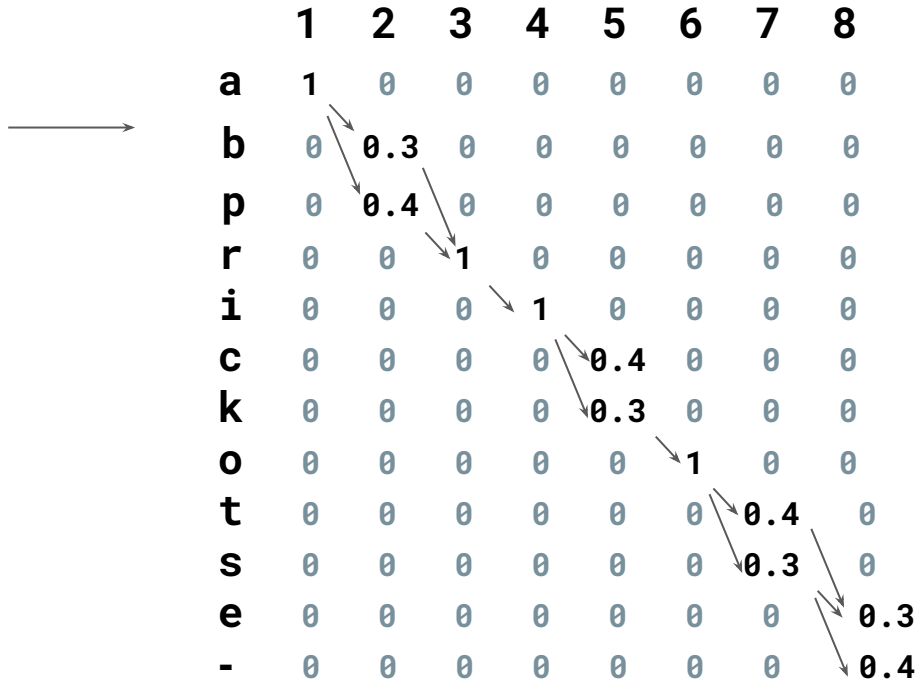
=> better sensitivity, but can be improved : HMMs

apricot-
abricot-
aprikose

Further than blast : sequence-profile comparison

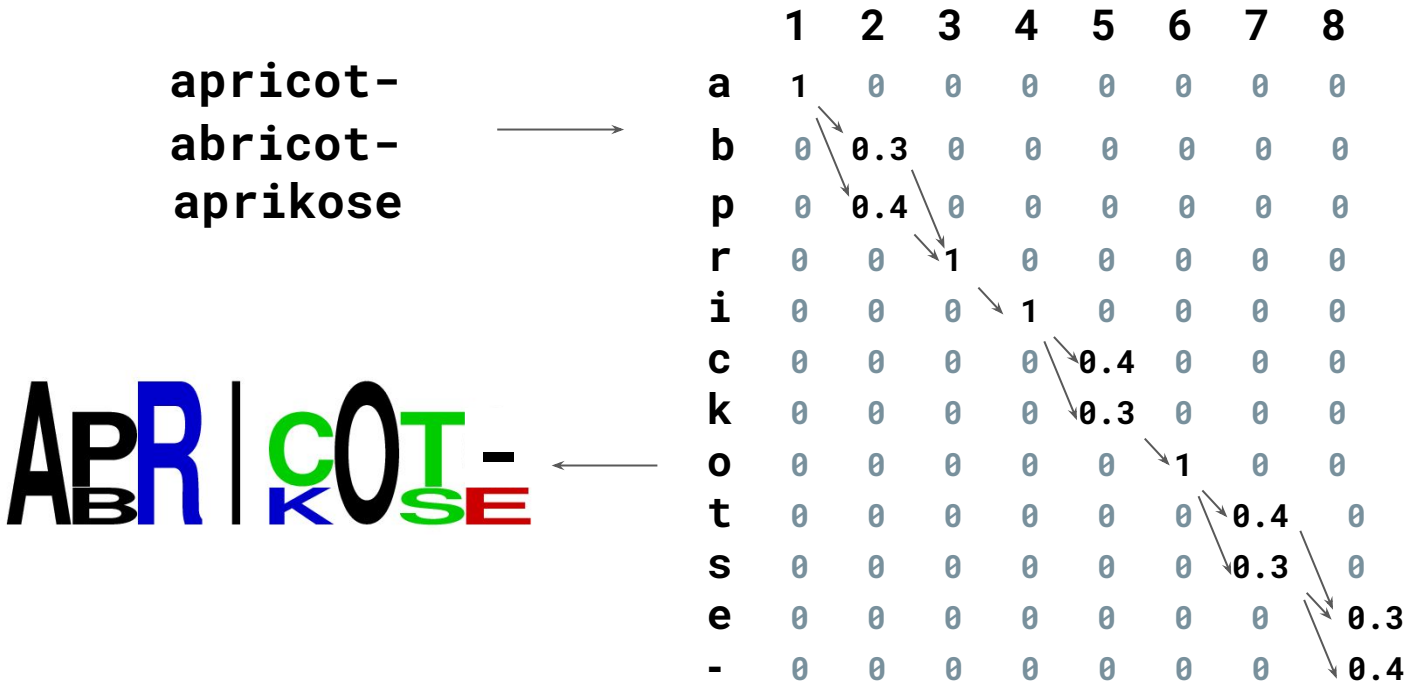
=> better sensitivity, but can be improved : HMMs

apricot-
abricot-
aprikose



Further than blast : sequence-profile comparison

=> better sensitivity, but can be improved : HMMs



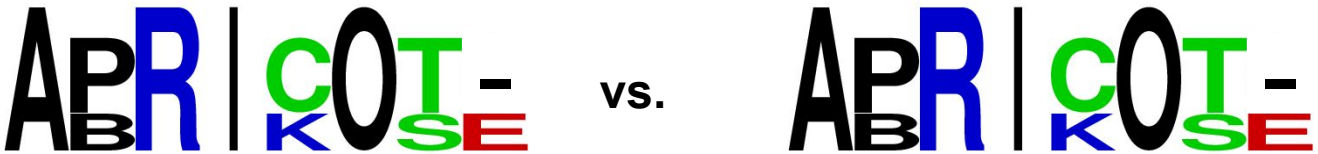
Further than blast : sequence-profile comparison

Compare a sequence to a HMM (HMMER)
“sequence-profile” comparison

apricot vs

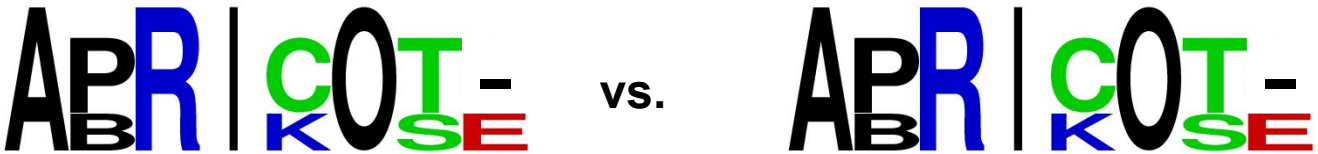
Further than sequence-profile : profile-profile comparison

HMM-HMM comparison **:** hhpred (includes secondary structure information) and hhblits (faster, iterative)



Further than sequence-profile : profile-profile comparison

HMM-HMM comparison **:** hhpred (includes secondary structure information) and hhblits (faster, iterative)



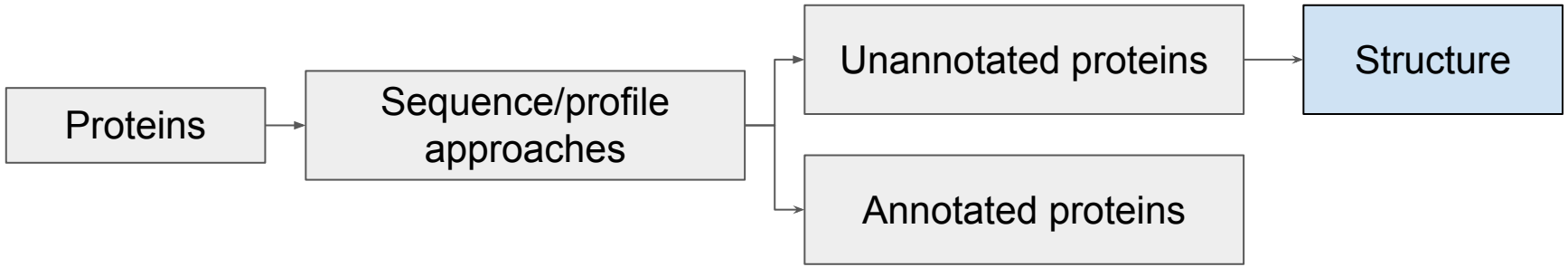
This is the best of what we can accomplish with sequences.

But protein sequences can diverge so much that we reach <10% sequence identity.

And even further : structures

To go further :
Structures can be more conserved and make the homology possible to detect

Proteins with identical structures have the same function



Structure prediction

First step is prediction structure.
Before : crystallography, NMR
-> But very long and complex.

Structure prediction

First step is prediction structure.
Before : crystallography, NMR
-> But very long and complex.

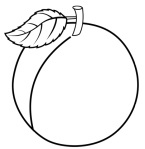
Now, Alphafold, using AI and deep learning : gold standard of *de novo* prediction of protein 3D structure from their amino acid sequence.

Structure prediction

First step is prediction structure.
Before : crystallography, NMR
-> But very long and complex.

Now, Alphafold, using AI and deep learning : gold standard of *de novo* prediction of protein 3D structure from their amino acid sequence.

Alphafold

apricot → 

aprikose → 

Structure prediction

A small window on AI in bioinformatics :

Alphafold uses deep learning : giving a lot of information to an AI to learn “by itself” the underlying functioning of something

Structure prediction

A small window on AI in bioinformatics :

Alphafold uses deep learning : giving a lot of information to an AI to learn “by itself” the underlying functioning of something

But : it’s a black box

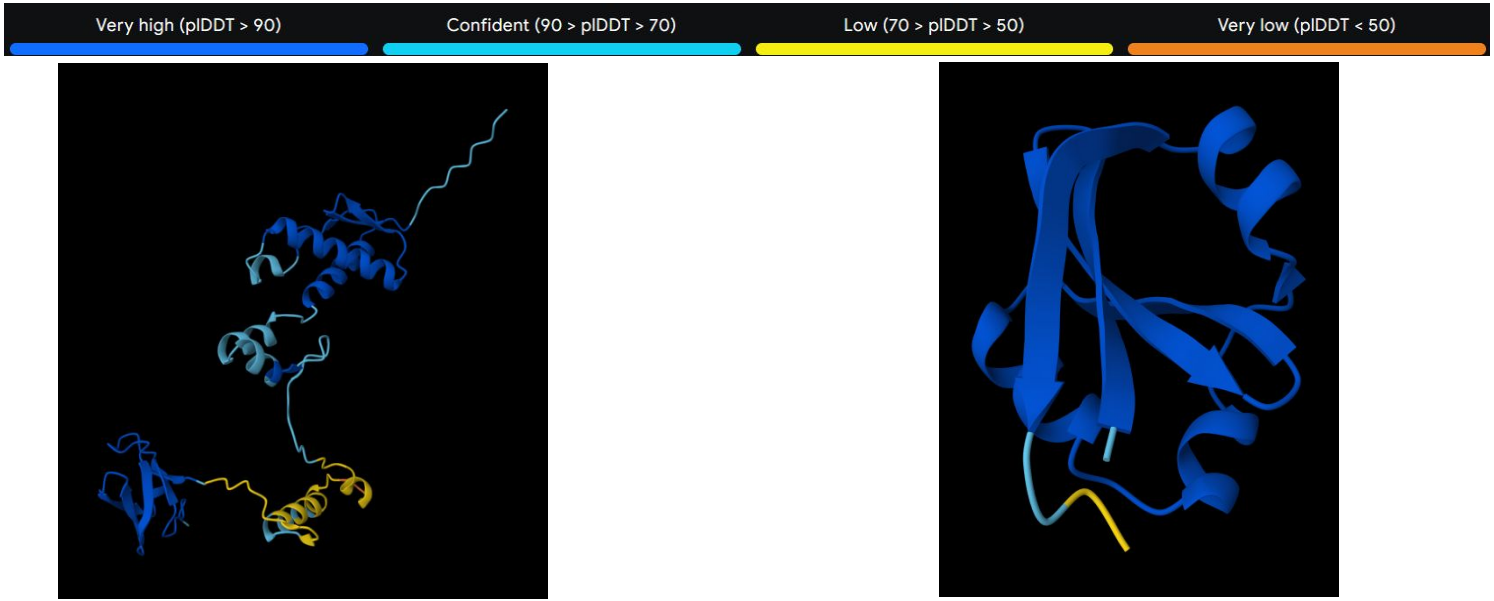
What’s in there ? Models, neural networks, vectors…
In the end, no “graspable” biologic reality. But it works !

Structure prediction

Prediction = imperfect, not all proteins can have a well predicted structure.
pLDDT (predicted Local Distance Difference Test) is a measure of certainty per position.

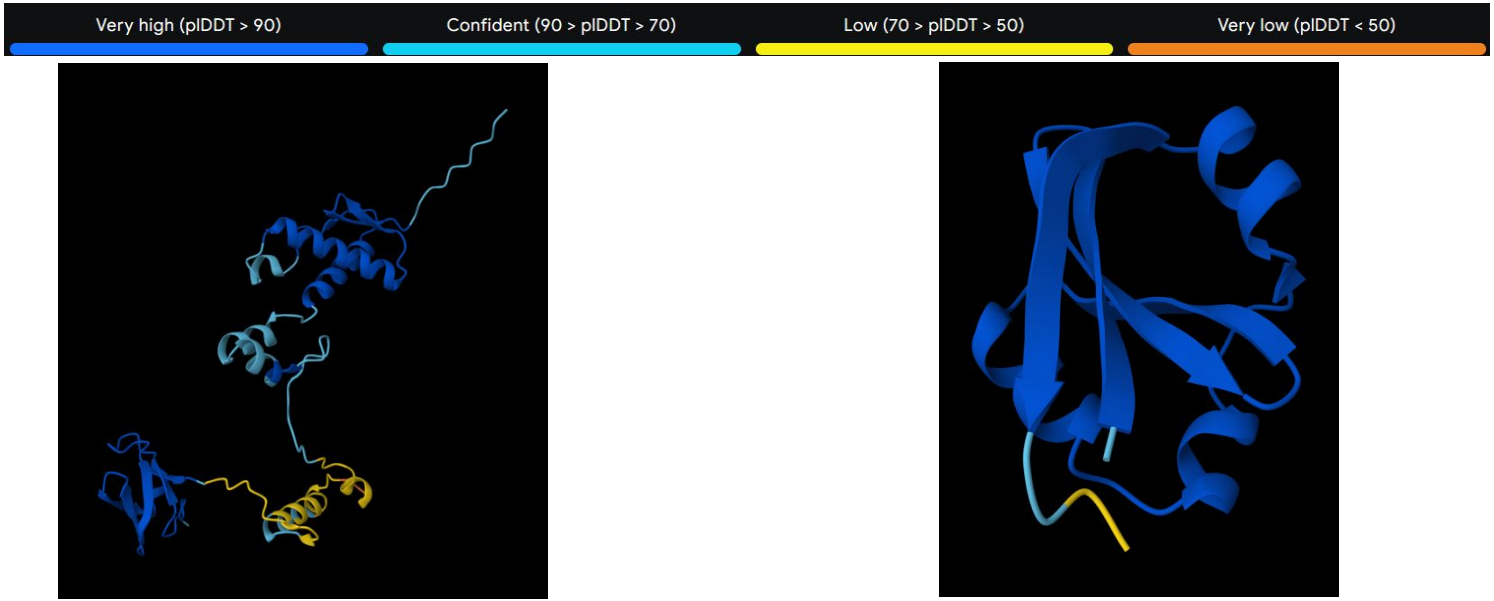
Structure prediction

Prediction = imperfect, not all proteins can have a well predicted structure.
pLDDT (predicted Local Distance Difference Test) is a measure of certainty per position.



Structure prediction

Prediction = imperfect, not all proteins can have a well predicted structure.
pLDDT (predicted Local Distance Difference Test) is a measure of certainty per position.



Average pLDDT > 0.7 is an acceptable score.

An history of structure comparison

Structure comparison tools since 1990 : Dali, CE…
But comparing 3D structures is computationally very intensive

An history of structure comparison

Structure comparison tools since 1990 : Dali, CE…
But comparing 3D structures is computationally very intensive

=> all of these softwares are very slow, not suited for large dataset comparison

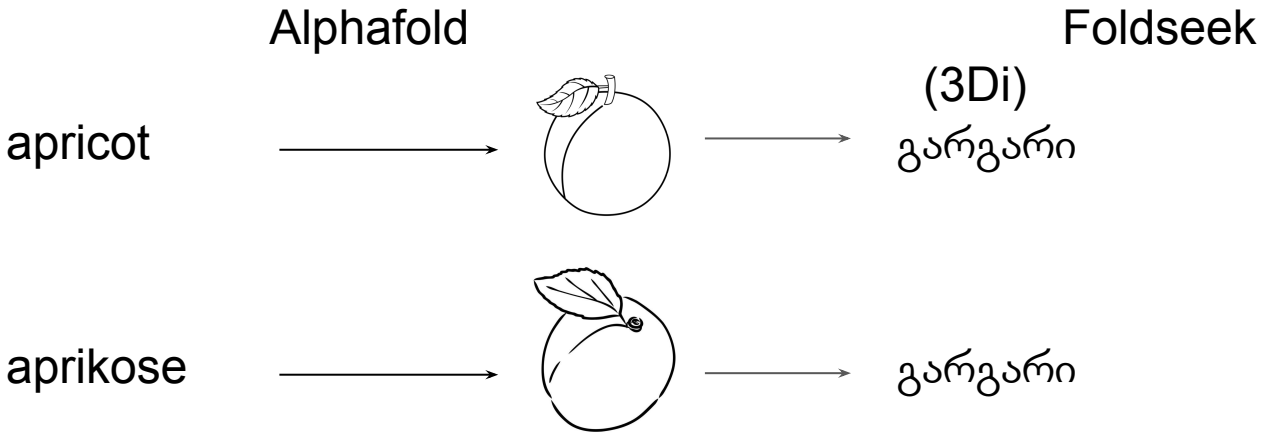
Structure comparison now

A revolution with Foldseek.
-> a new alphabet, called 3Di, is coded describing the relative spatial position of each amino acid before and after the amino acid in question.

Adapted from
Van Kempen et al. 2023

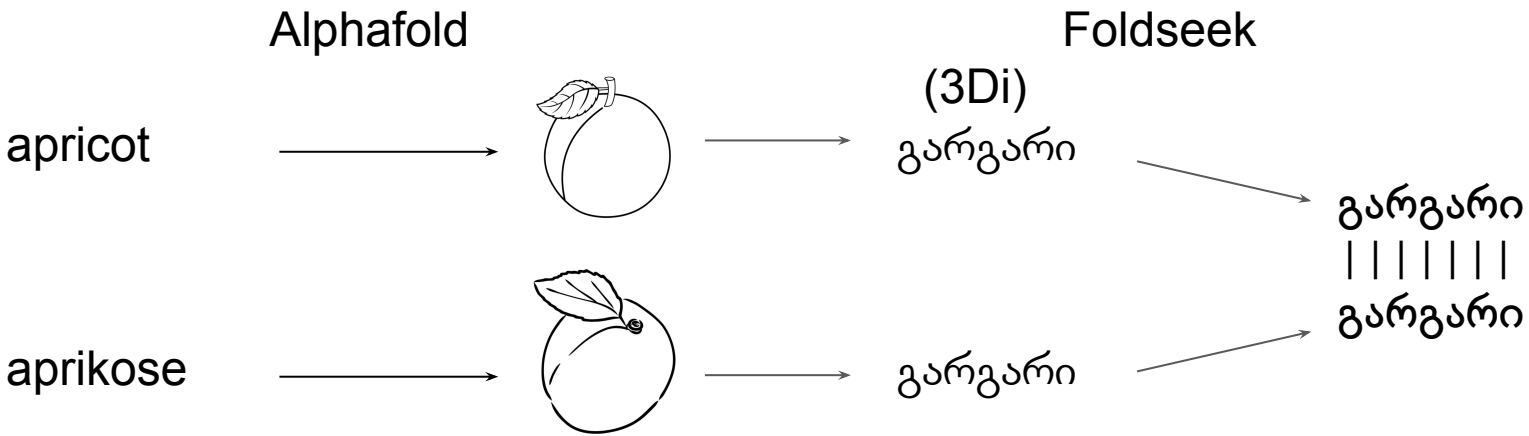
Structure comparison now

A revolution with Foldseek.
-> a new alphabet, called 3Di, is coded describing the relative spatial position of each amino acid before and after the amino acid in question.



Structure comparison now

A revolution with Foldseek.
-> a new alphabet, called 3Di, is coded describing the relative spatial position of each amino acid before and after the amino acid in question.
Making coded sequences which can then be compared



Comparing… but against what ?

We now have multiple comparison approach, more or less sensitive and computationally intensive.

What we want for a clean phage genome annotation :
i) a database of annotated phage proteins
ii) to transfer their annotation to proteins of our genome using different methods

Comparing… but against what ?

We now have multiple comparison approach, more or less sensitive and computationally intensive.

What we want for a clean phage genome annotation :
i) a database of annotated phage proteins
ii) to transfer their annotation to proteins of our genome using different methods

PHROGs

i) PHROGs : the best available database of this type.
A clustered database of prokaryotic virus protein families.

Clustered proteins -> as many proteins as possible per family (called “phrog”)

phrog_514 is made of **265** protein sequences and is annotated as "**major head protein**"
Functional category: **head and packaging**

-> an MSA was built with proteins of this family
-> an HMM was built from this MSA

Comparing… but against what ?

We now have multiple comparison approach, more or less sensitive and computationally intensive.

What we want for a clean phage genome annotation :
i) a database of annotated phage proteins
ii) to transfer their annotation to proteins of our genome using different methods

Using PHROGs to annotate

i) it is possible to compare our proteins to all proteins of PHROGs, (sequence-sequence comparison, BLAST)

Using PHROGs to annotate

i) it is possible to compare our proteins to all proteins of PHROGs, (sequence-sequence comparison, BLAST)

ii) compare our proteins to all HMMs of PHROGs (sequence-profile comparison, HMMER)

Using PHROGs to annotate

i) it is possible to compare our proteins to all proteins of PHROGs, (sequence-sequence comparison, BLAST)

ii) compare our proteins to all HMMs of PHROGs (sequence-profile comparison, HMMER)

iii) to build an MSA for each of our proteins using external sequences, and compare these to the PHROG profiles with HHpred (profile-profile comparison)

Using PHROGs to annotate

i) it is possible to compare our proteins to all proteins of PHROGs, (sequence-sequence comparison, BLAST)

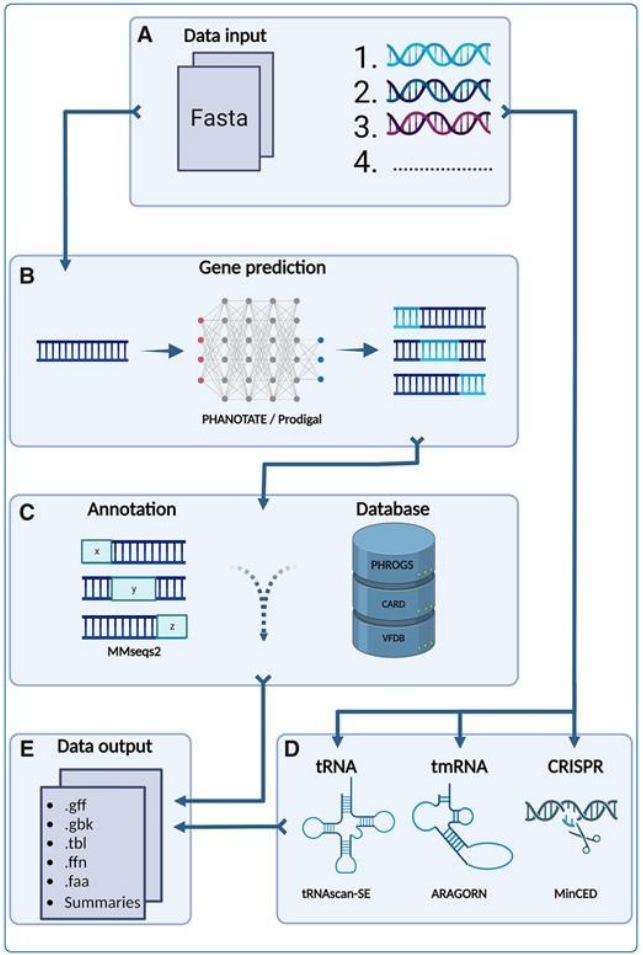
ii) compare our proteins to all HMMs of PHROGs (sequence-profile comparison, HMMER)

iii) to build an MSA for each of our proteins using external sequences, and compare these to the PHROG profiles with HHpred (profile-profile comparison)

iv) comparing structures of our proteins to structures of PHROGs (structure-structure comparison)

Pharokka, Phold : automatic workflows

An automatic tool : Pharokka
A sequence-profile comparison to the PHROG database
Available on Galaxy

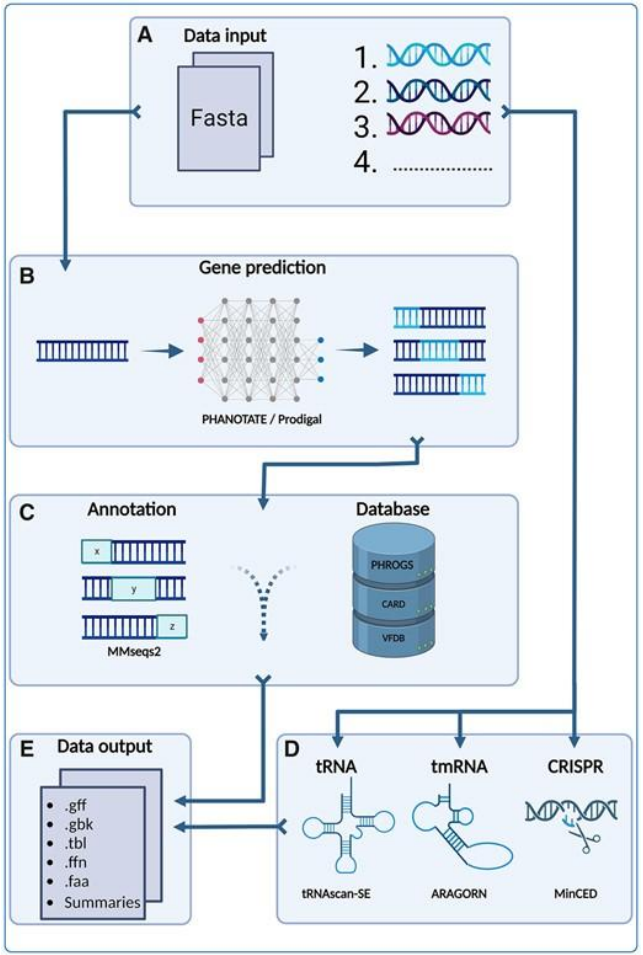


Bouras et al. 2023

Pharokka, Phold : automatic workflows

An automatic tool : Pharokka
A sequence-profile comparison to the PHROG database
Available on Galaxy

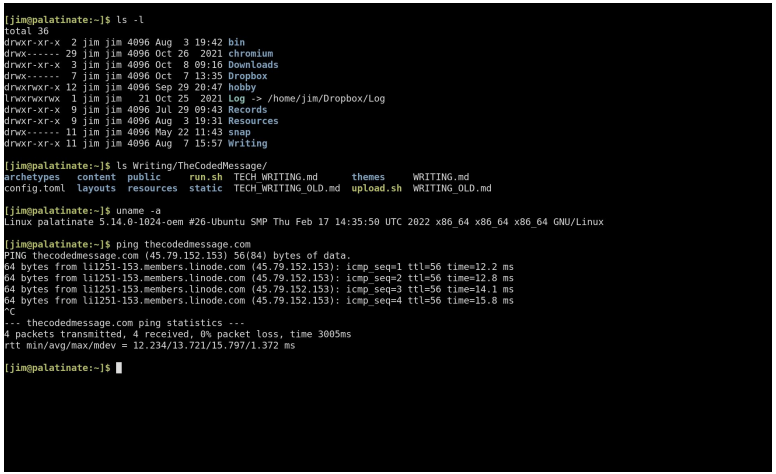
Another tool : Phold
Predicting protein structure and comparing them to structures of PHROGs
Available on Command lines



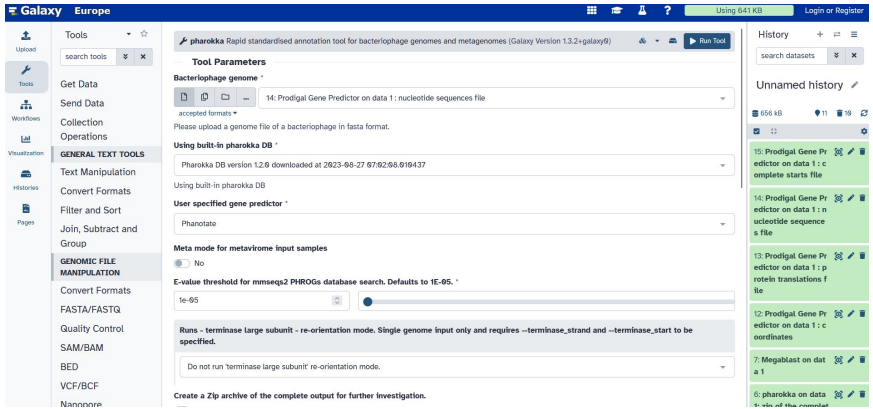
Bouras et al. 2023

Which working environment ?

This afternoon, we'll use galaxy for pharokka :
An open source scientific workflow with an easy to use web interface.



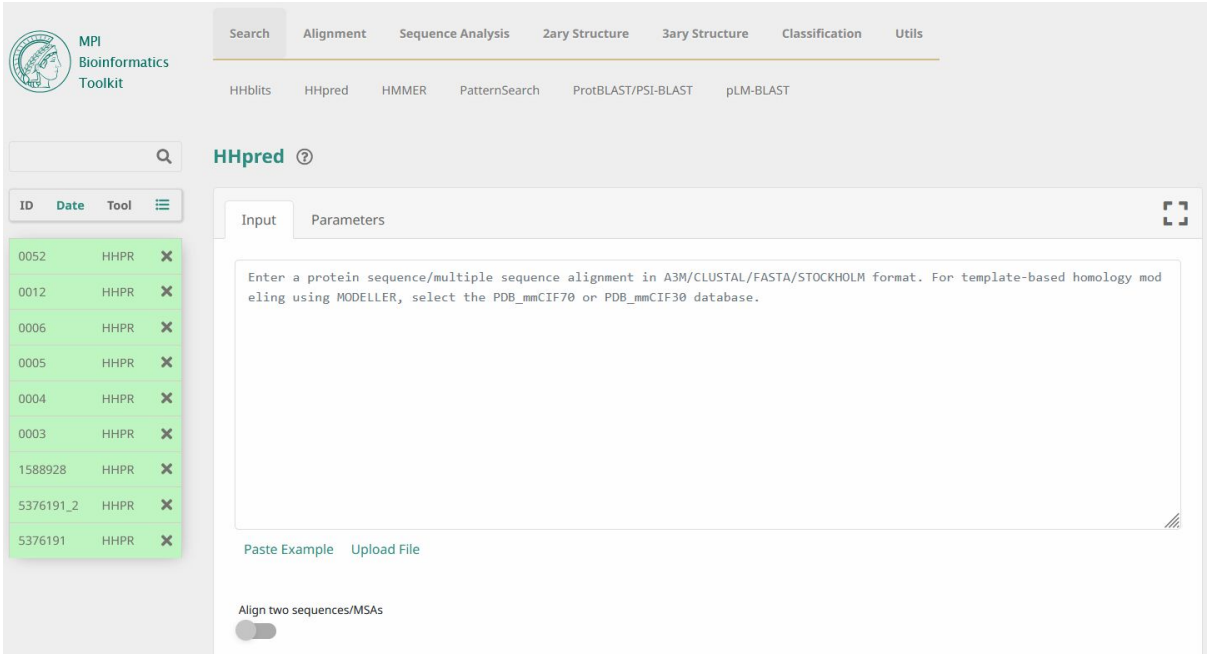
Command lines : too complicated, not enough time today



Galaxy : easy, clear, accessible

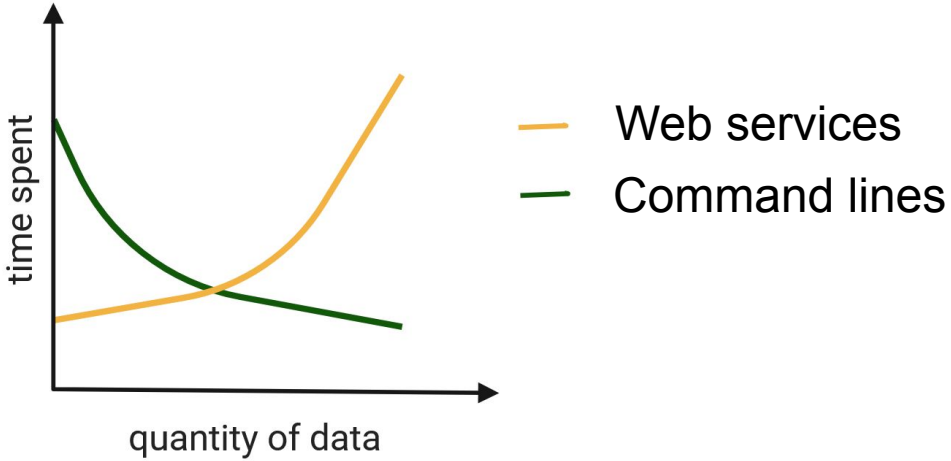
Which working environment ?

Likewise, we'll use HHpred on MPI bioinformatics toolkit :
A website with a lot of bioinformatic tools



Going further

This was just a small introduction of a lot of concepts.



For bigger analysis, learning rudiments of command lines is necessary

Take home

-> Protein annotation is done by homology detection

Take home

-> Protein annotation is done by homology detection

-> There are multiple homology detection methods

Take home

-> Protein annotation is done by homology detection

-> There are multiple homology detection methods

-> There are a lot of trade-offs in bioinformatics

Take home

-> Protein annotation is done by homology detection

-> There are multiple homology detection methods

-> There are a lot of trade-offs in bioinformatics

-> PHROGs is the best database for clean phage protein annotations

Thank you for your attention

mail : quentin.nugier@inrae.fr