Alignment

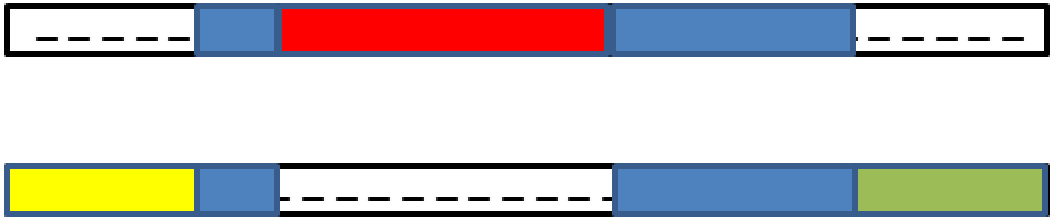
Importance of alignment

- The most important basic question about a gene or protein is ***whether it is related*** to any other gene or protein!
- Relatedness for two proteins suggests:
  - That they are homologous
  - They may have a common function
- Analysis of DNA and protein sequences identifies domains or motifs that are shared among a group of molecules.
- Analysis is accomplished by ***Sequence alignment***.
- Protein alignment is more informative than DNA alignment.

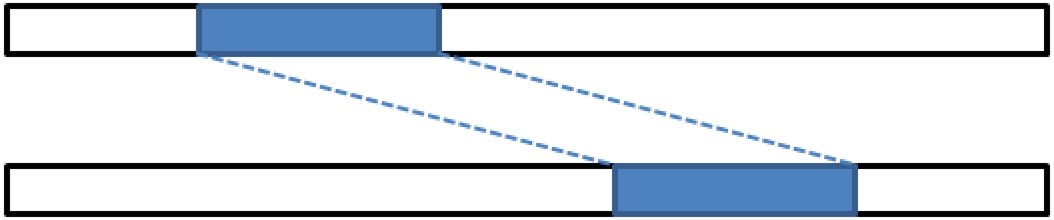
Types of Alignment

- 1- Global Alignment:
  - Aligning the ***entire*** length of two sequences.
- 2- Pairwise (local) alignment:
  - Aligning ***part*** of the sequence with an ***entire*** length.
  - A subset of the two sequences are aligned.

Types of Alignment



Global Alignment



Local Alignment

Definitions

- **Homology:**
  - It is the state of having the same or similar relation, relative position or structure
  - Homologous sequences share a common evolutionary ancestry
  - Two homologous sequences (either amino acid or nucleotide sequences) usually share significant identity
  - Two types of homologous proteins:
    - Orthologous:
      - Are homologous sequences in different species that arose from a common ancestral gene during speciation
      - Have similar biological functions
    - Paralogous:
      - Are homologous sequences that arose by a mechanism such as gene duplication
  - Definition of homology is based on alignment scores
  - Homologous ≠ same function

Definitions

- **Identity:**
  - Is the extent to which two amino acid (or nucleotide) sequences are invariant
- **Similarity:**
  - Aligned residues are similar but not identical
  - Share similar biochemical properties
  - Similar pairs are structurally or functionally related

How do you align sequences?

- Visually? ….. NO! very difficult!
- Computer algorithm? …..YES!!

Introduction to sequence alignment

- **Hamming Distance:**
  - Counts mismatches in two strings
  - Assumes we align the *ith* symbol in the first sequence to the *ith* symbol in the 2nd sequence.

Example: Compute the hamming distance?

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

ZERO Matches!!!

hamming distance=8

But…

- If we **align** the sequences differently you'll have six **matching** positions

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

SIX Matches!!!

Good alignment?

Alignment 1

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

Alignment 2

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

- The alignment that matches as many symbols as possible is the good alignment.

The alignment game

Example:

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

Alignment Game (maximizing the number of points):

- Remove the 1st symbol from each sequence
  - 1 point if the symbols match, 0 points if they don’t match
- Remove the 1st symbol from one of the sequences
  - 0 points

The alignment game

```
 A T G T T A T A
 A T C G T C C
+1
```

The alignment game

```
 A T G T T A T A
 A T C G T C C
+1+1
```

The alignment game

```
 A T - G T T A T A
 A T C G T C C
+1+1
```

The alignment game

```
 A T - G T T A T A
 A T C G T C C
+1+1  +1
```

The alignment game

```
 A T - G T T A T A
 A T C G T C C
+1+1  +1+1
```

The alignment game

```
 A T - G T T A T A
 A T C G T - C C
+1+1  +1+1
```

The alignment game

```
 A T - G T T A T A
 A T C G T - C C
+1+1  +1+1
```

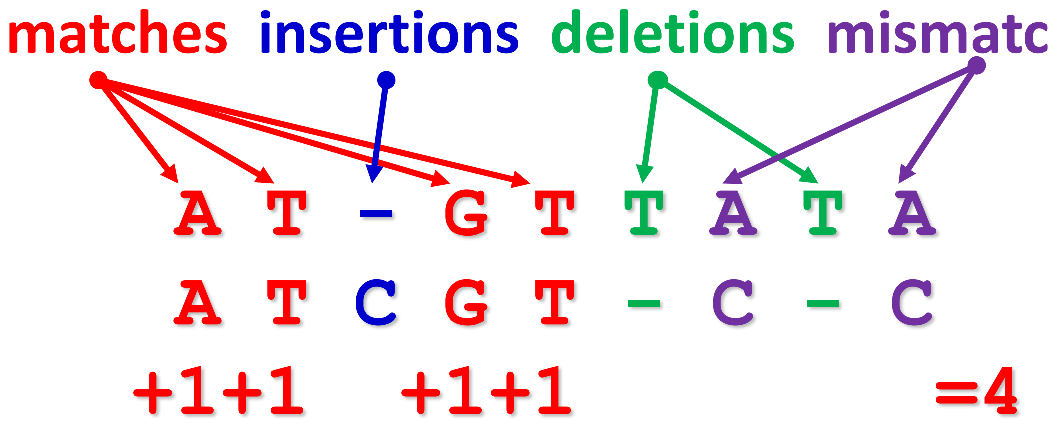
The alignment game

```
 A T - G T T A T A
 A T C G T - C - C
+1+1  +1+1
```

The alignment game

```
 A T - G T T A T A
 A T C G T - C - C
+1+1  +1+1        =4
```

What is the sequence alignment?



matches insertions deletions mismatches

```
A T - G T T A T A
A T C G T - C - C
+1+1 +1+1       =4
```

Alignment of two sequences is a two-row matrix:

1st row: symbols of the 1st sequence (in order) interspersed by “-”

2nd row: symbols of the 2nd sequence (in order) interspersed by “-”

We can see that letters may:

Match: The two letters are the same

Mismatch: The two letters are different

Indel (INsertion or DELetion): One letter aligns to a ***gap*** *in the other string.*

Alignment

An **alignment** of sequences *"v"* and *"w"*:

- a two-row matrix
- such that the first row contains the symbols of v in order
- the second row contains the symbols of w in order
- space symbols may be interspersed throughout each string.
- Two space symbols are not aligned against each other.

Longest Common Subsequence

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

Matches in alignment of two sequences (**ATGT**) form their **Common Subsequence**

Longest Common Subsequence Problem: Find a longest common subsequence of two strings.

- **Input:** Two strings.
- **Output:** A longest common subsequence of these strings.

Is this a useful alignment

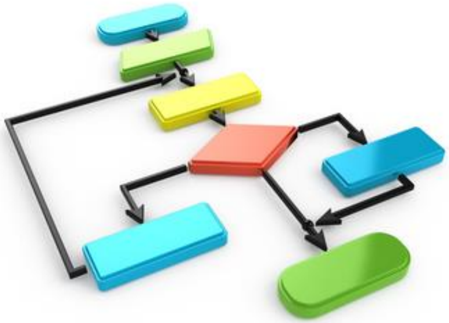
```
Seq1      GGGAATGCGTAGCATCGA
Seq2      GGCACTGATCGATGCTACG
```

```
Seq1      GGG----AAT------GCGTAGC----AT------CGA
Seq2      ---GGCA---CTGATC--------GATG--CTACG---
```

- What will happen if aligning two sequences with different length.
- The answer is to introduce gaps in the shortest sequence
- The alignment with highest score is optimum!!!

Summary

- Pairwise alignment is the process of lining up two sequences to achieve ***maximal levels identity***.



What is an algorithm?

An algorithm is a procedure or formula for solving a problem. Developed by Mohammed ibn-Musa al-Khwarizmi (201H – 271H).

Global alignment optimum algorithm

- It is also called **Needleman-Wunsch** algorithm.
- *Also used in Google search engine!*
- Used to calculate the **optimum** alignment (means the maximum score = good alignment).
- It is a kind of **Dynamic programming**. Solving large problem by dividing it to small problems.
- It is composed of three steps:
  - initiation
  - Filling
  - Trace-back
- Align these two sequences: CGCA & CACGTAT

Step 1: Initiation

- Design a scoring metric (these numbers vary and you can set your own scoring metrics):

Match =	**1**
Mismatch =	**0**
Gap (indel) penalty =	**-1**

Step 1: Initiation

Make a matrix and add gap for each sequence

The score of the alignment would be 0

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

Step 2: Iteration (filling the matrix)

Each cell has three possibilities:

- To introduce a gap horizontally (in the first seq).
- To introduce a gap vertically (in the second seq).
- To calculate if they match or mismatch and add to the diagonal cell.

The highest score is added and recorded the direction from which cell it came.

- MEANING .. the cell has three possible candidate sums:

➢ The top neighbor has score -1 and moving from there represents an indel, so add the score for indel: (-1) + (-1) = (-2)

➢ The left neighbor also has score -1, represents an indel and also produces (-2).

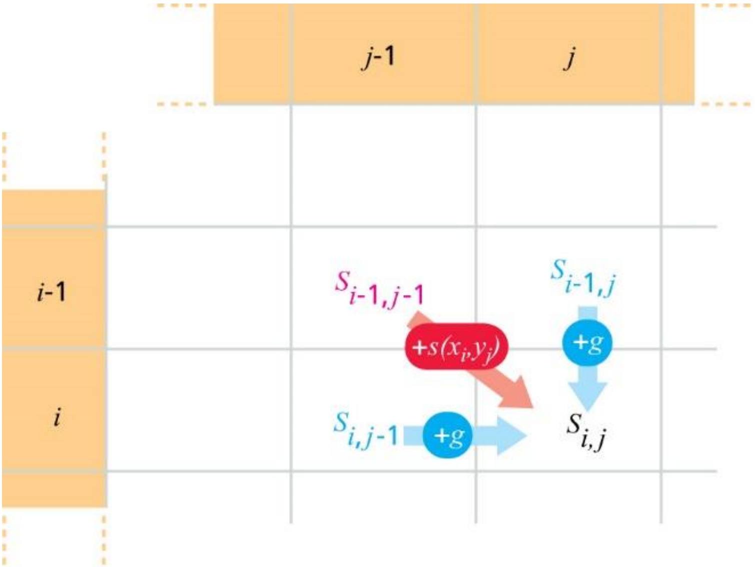
➢ The diagonal top-left neighbor has score 0. The pairing of C and C is a match, so add the score for match: 0+1 = 1

➢ *The highest candidate is 1 and is entered into the cell*

$s_{i-1,\, j}$ + weight of edge "↓" into (*i,j*)

$s_{i,\, j-1}$ + weight of edge "→" into (*i,j*)

$s_{i-1,\, j-1}$+ weight of edge "↘" into (*i,j*)



Step 2: Iteration (filling the matrix)

-1 + (-1) = -2

-1 + (-1) = -2

0 + (1) = 1

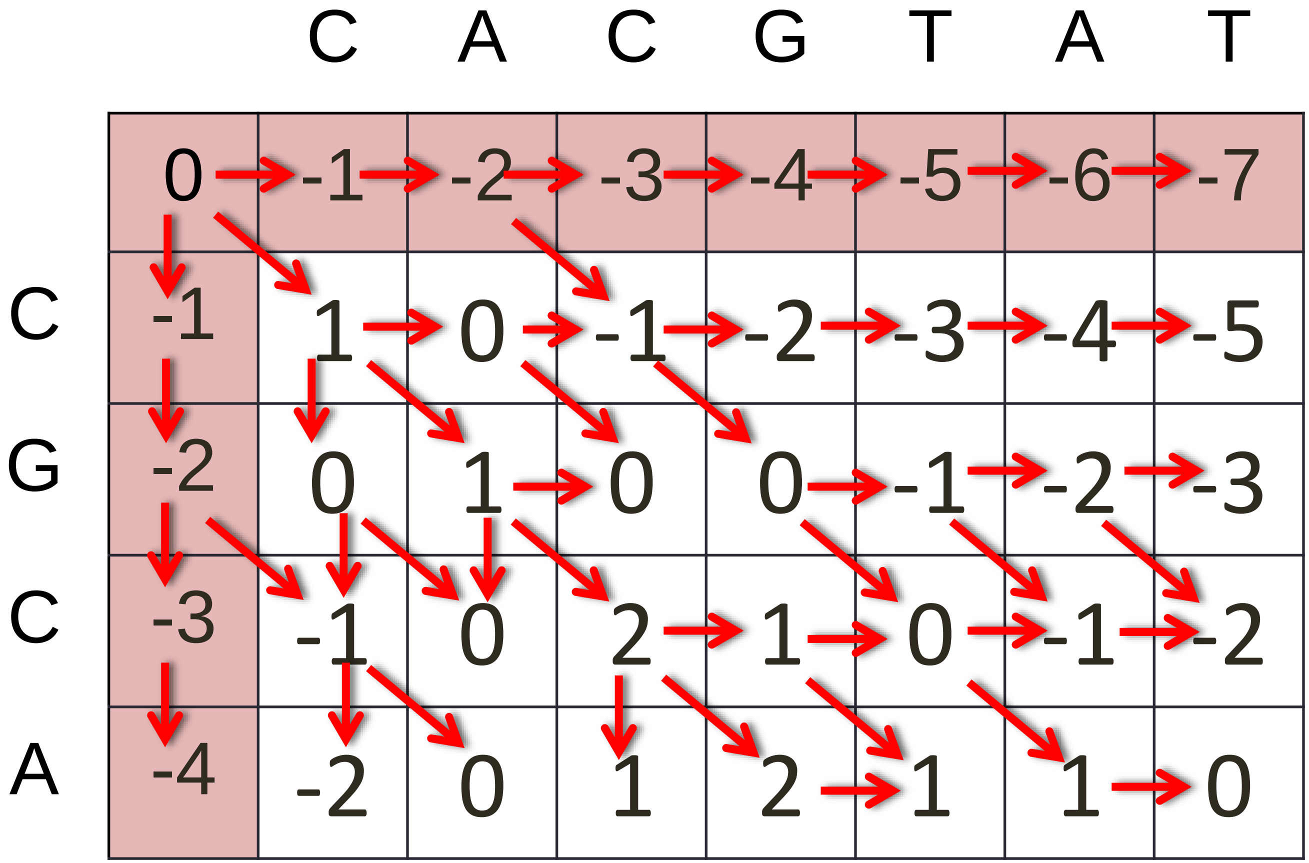
- -2, -2, 1

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

Match = 1 | Mismatch = 0 | Gap (indel) penalty = -1

Step 2: Iteration (filling the matrix)

Match = 1 | Mismatch = 0 | Gap penalty = -1



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

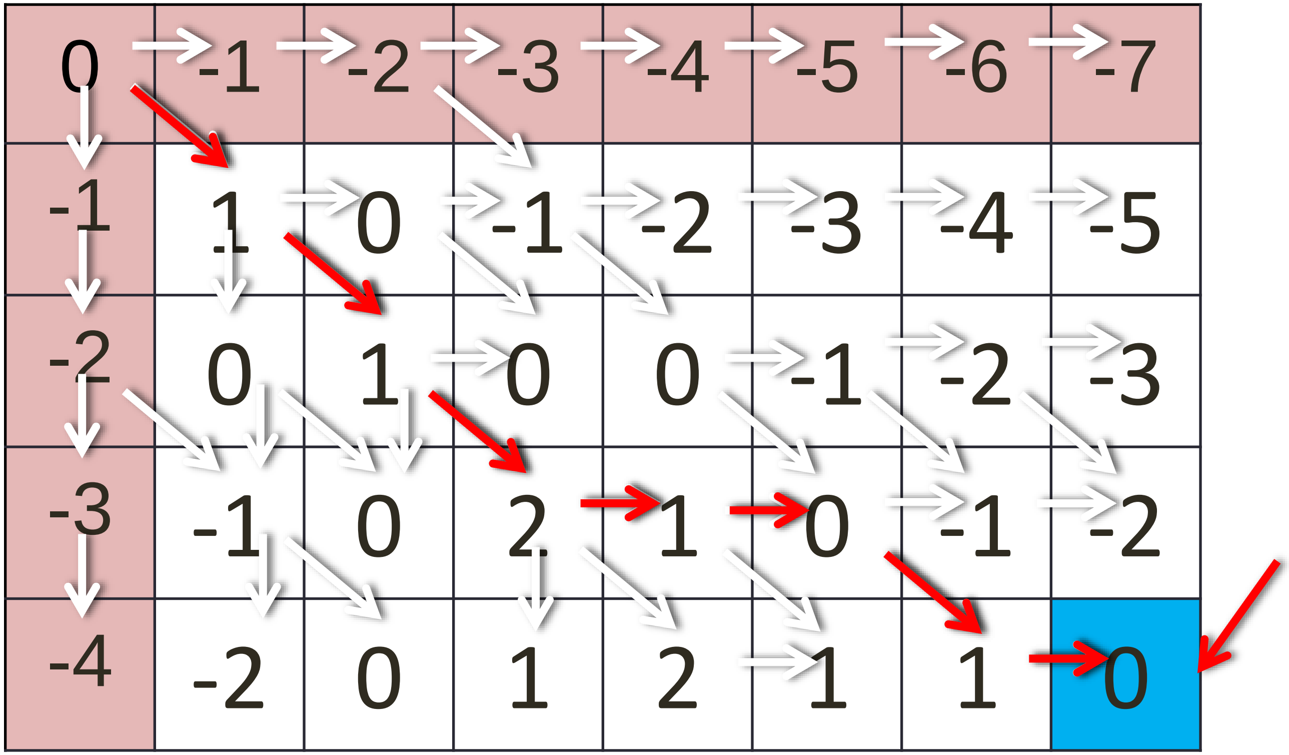
Step 3: Trace-back rules

- Start from the bottom right corner of the square.
- Add gap in the **first** (horizontal) sequence if arrows are located **horizontally**.
- Add gap in the **second** (vertical) sequence if arrows are located **vertically**.
- Align the two sequences if the arrow is diagonal.

Step 3: Trace-back

CACGTAT

CGC--A-



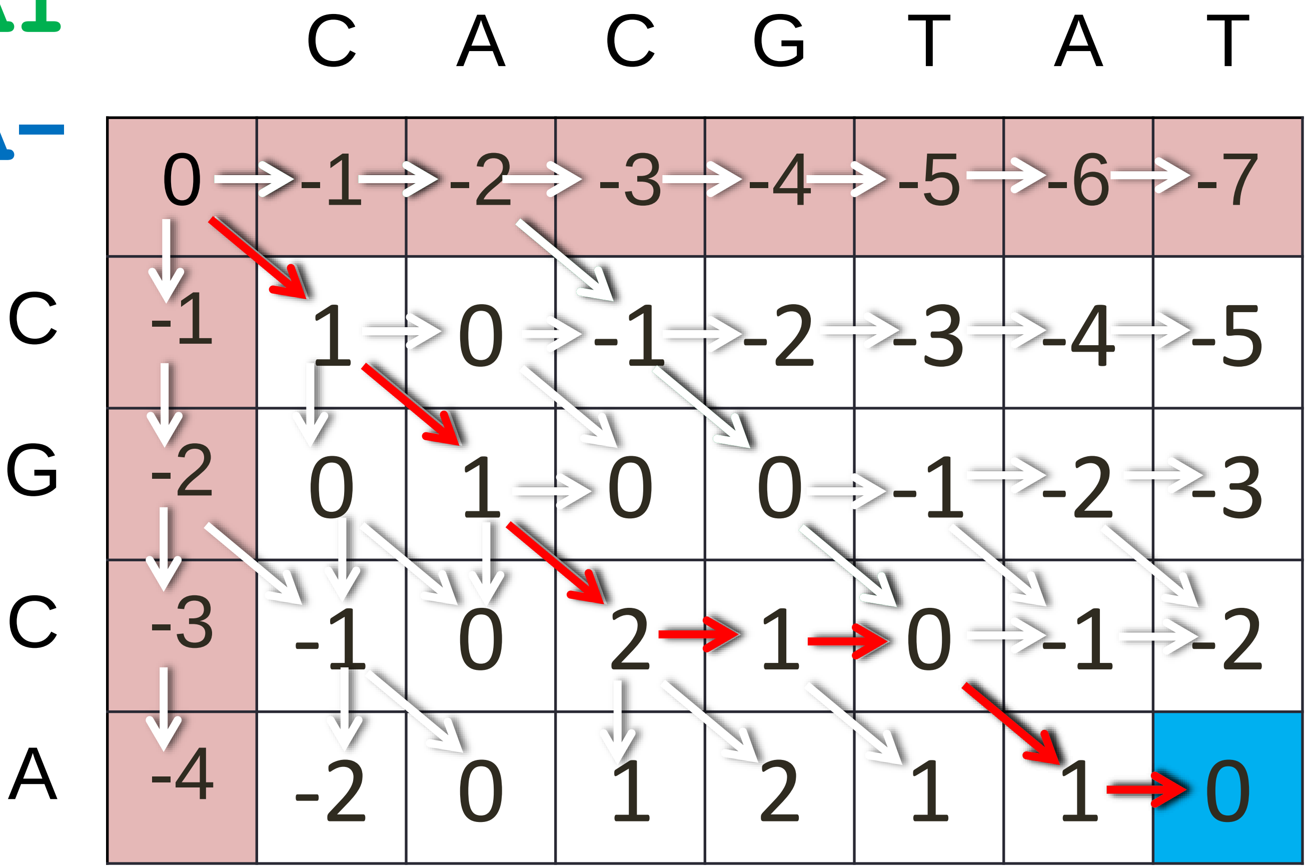
		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

Start here

Step 3: Trace-back

(A Second answer)

```
CACGTAT
--CGCA-
```

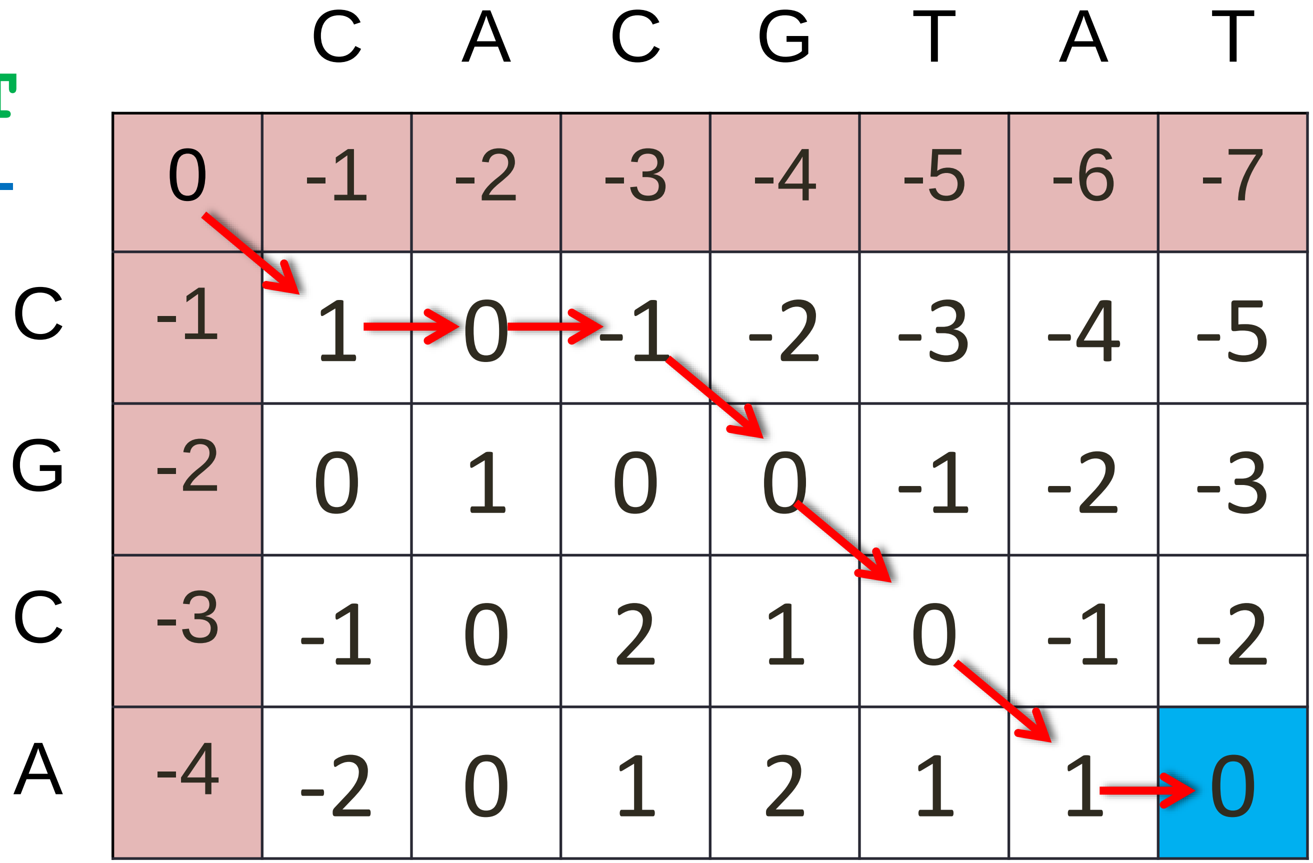


		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

Step 3: Trace-back

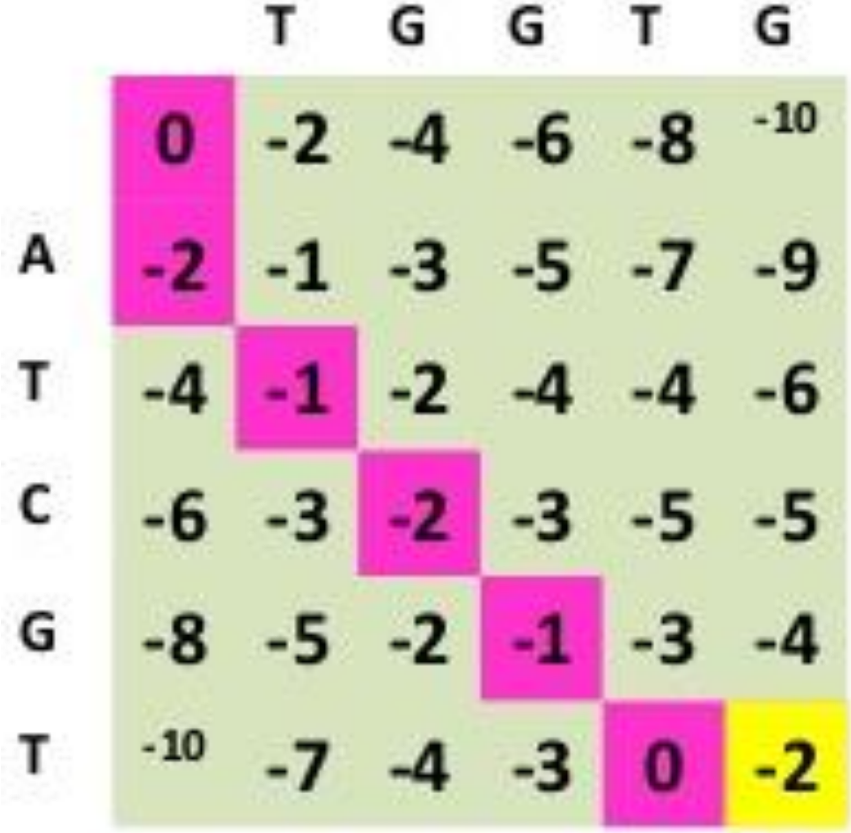
(A Third answer)

CACGTAT
C--GCA-



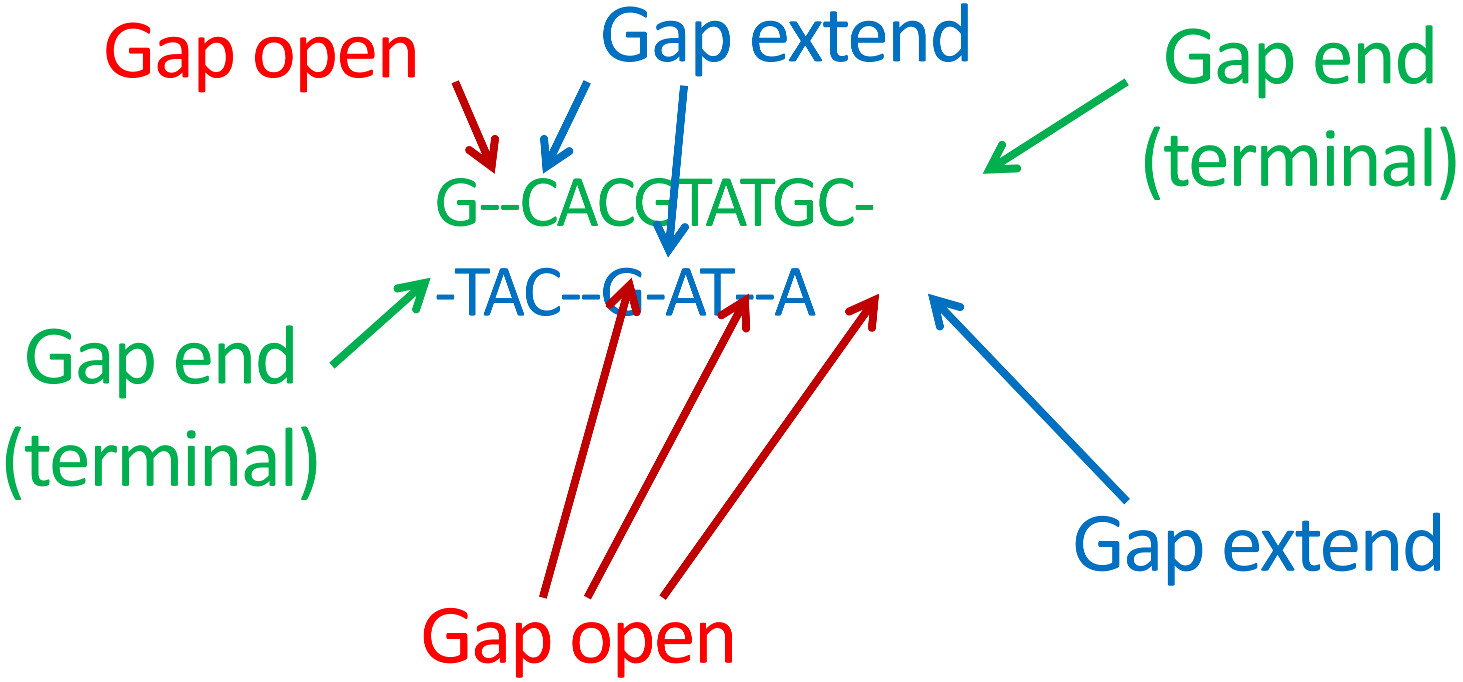
		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

Deduce the alignment



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

Different gap penalty meaning



Terminal gaps is preferred over gap introduction.

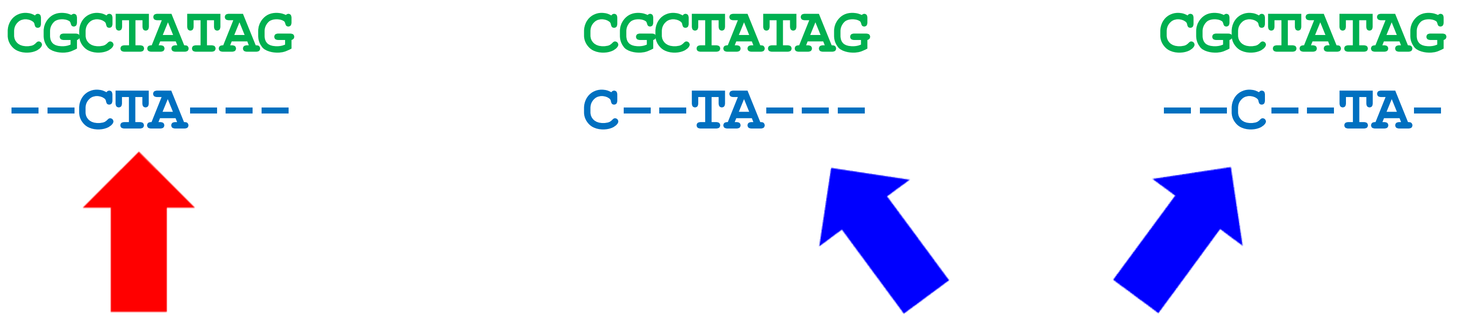
Gap penalty value could change

- When comparing two protein coding genes, then penalizes gap high because of the frameshift problem.
- When comparing genes for noncoding RNA, we could set gap penalty lower (because gap is worse than mismatch).

- If you search for sequences that are strict match to your query, then set the penalty gap to high value.
- If you search for similarity between distantly related sequences, then set gap penalty to low value.

Local alignment

(Smith-Waterman Algorithm)



```
CGCTATAG      CGCTATAG      CGCTATAG
--CTA---      C--TA---      --C--TA-
```

Local Alignment

Global Alignment

Why using local alignment (Smith-Waterman Algorithm)

- It allow searching for certain sequences within large sequence.
- To identify pattern within protein sequence
- To identify transcription binding site
- To identify regulatory elements within a genome
- Local alignment looks for optimal partial (subsequence) matches.

Roles for local alignment

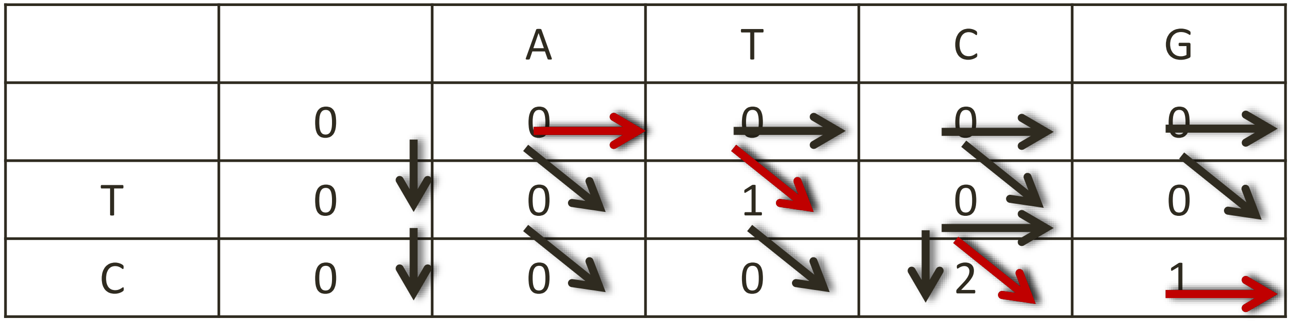
- It is exactly as Needleman-Wunsch Algorithm
- *Negative value* is replaced by zero (0).
- Align these two sequences using Smith-Waterman algorithm. ATCG & TC

Match	= 1
Mismatch	= 0
Gap (indel) penalty	= -1

Align these two sequences using Smith-Waterman algorithm

Match	= 1
Mismatch	= 0
Gap (indel) penalty	= -1

```
ATCG
-TC-
```



		A	T	C	G
	0	0	0	0	0
T	0	0	1	0	0
C	0	0	0	2	1

Which Alignment is Better?

- Alignment 1: score = 22 (matches) - 20 (indels)=2.

```
GCC-C-AGT--TATGT-CAGGGGGCACG--A-GCATGCAGA-
GCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT
```

- Alignment 2: score = 17 (matches) - 30 (indels)=-13.

```
---G----C-----C--CAGTTATGTCAGGGGGCACGAGCATGCAGA
GCCGCCGTCGTTTTCAGCAGTTATGTCAG-----A------T-----
```

Which Alignment is Better?

- Alignment 1: score = 22 (matches) - 20 (indels)=2.

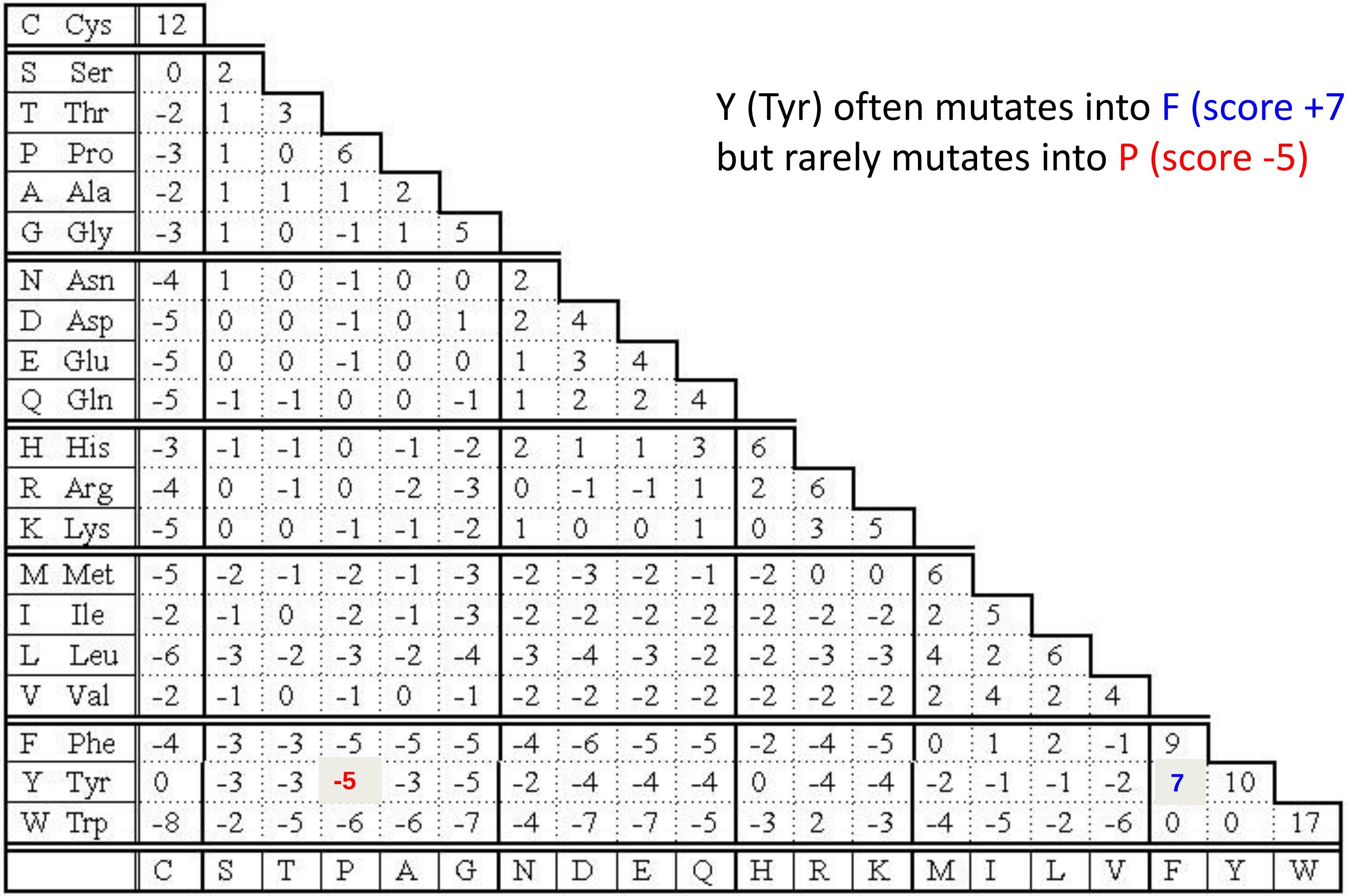
```
GCC-C-AGT--TATGT-CAGGGGGCACG--A-GCATGCAGA-
GCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT
```

- Alignment 2: score = 17 (matches) - 30 (indels)=-13.

```
---G----C-----C--CAGTTATGTCAGGGGGCACGAGCATGCAGA
GCCGCCGTCGTTTTCAGCAGTTATGTCAG-----A------T-----
```

local alignment

Scoring matrices for amino acid sequences



C	Cys	12																			
S	Ser	0	2																		
T	Thr	-2	1	3																	
P	Pro	-3	1	0	6																
A	Ala	-2	1	1	1	2															
G	Gly	-3	1	0	-1	1	5														
N	Asn	-4	1	0	-1	0	0	2													
D	Asp	-5	0	0	-1	0	1	2	4												
E	Glu	-5	0	0	-1	0	0	1	3	4											
Q	Gln	-5	-1	-1	0	0	-1	1	2	2	4										
H	His	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	Phe	-4	-3	-3	-5	-5	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Y (Tyr) often mutates into F (score +7)
but rarely mutates into P (score -5)

Scoring Gaps

- We previously assigned a fixed penalty *σ* to each indel.
- However, this fixed penalty may be too severe for a series of 100 consecutive indels.
- A series of *k* indels often represents a single evolutionary event (**gap**) rather than *k* events:

two gaps
(lower score)

```
GATCCAG
GA-C-AG
```

```
GATCCAG
GA--CAG
```

a single gap
(higher score)

From Pairwise to Multiple Alignment

- Up until now we have align two sequences only.
- 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.

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.

Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

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

Alignments = Paths in 3-D

- Alignment of ATGC, AATC, and ATGC

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

#symbols up to a given position

	A	A	T	--	C

	--	A	T	G	C

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

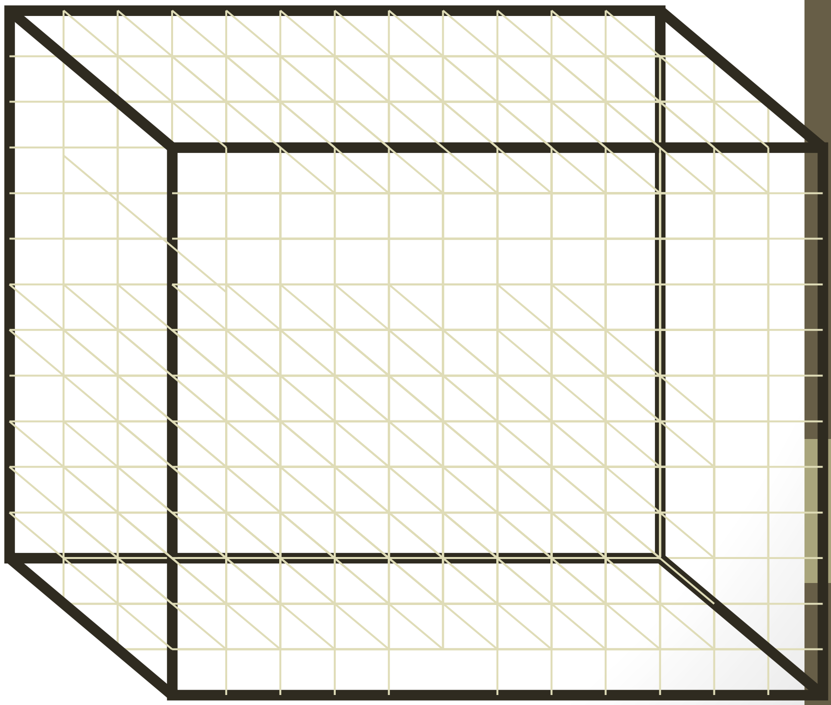
#symbols up to a given position

Alignments = Paths in 3-D

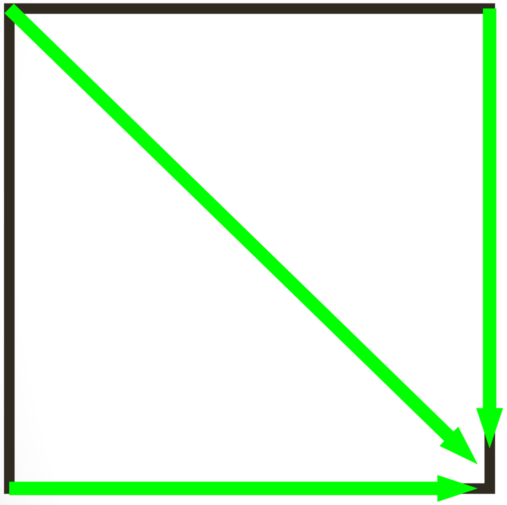
- Alignment of ATGC, AATC, and ATGC

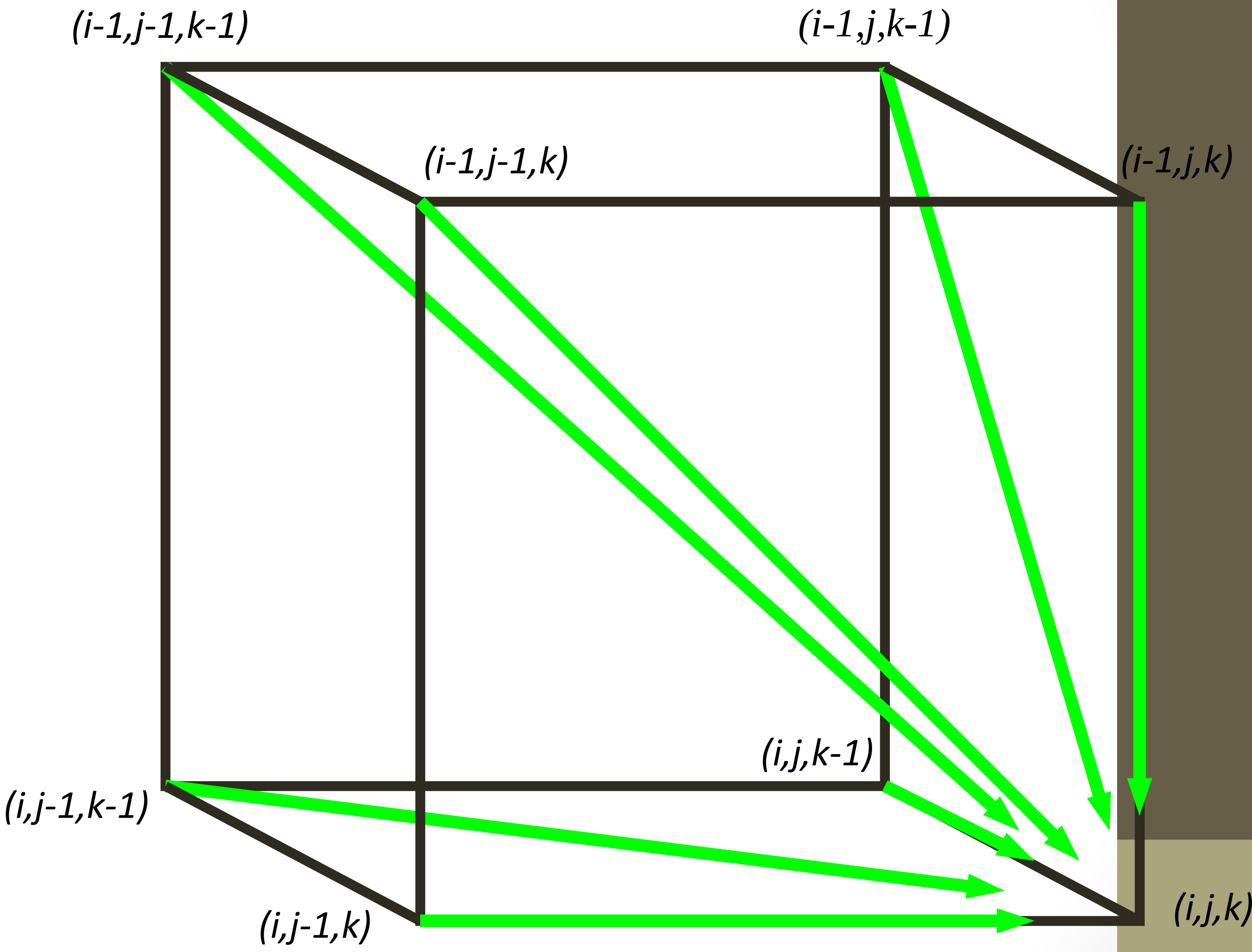
(0,0,0)→(1,1,0)→(1,2,1) →(2,3,2) →(3,3,3) →(4,4,4)

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



2-D Alignment Cell versus 3-D Alignment Cell

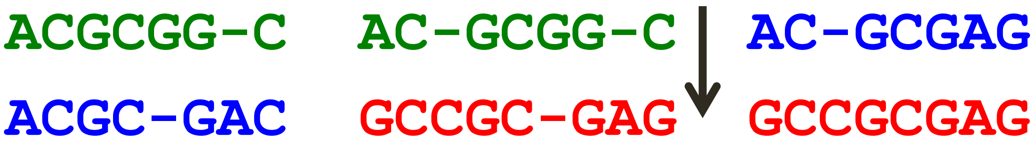

2-D



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
```

Homology

Types of Homology

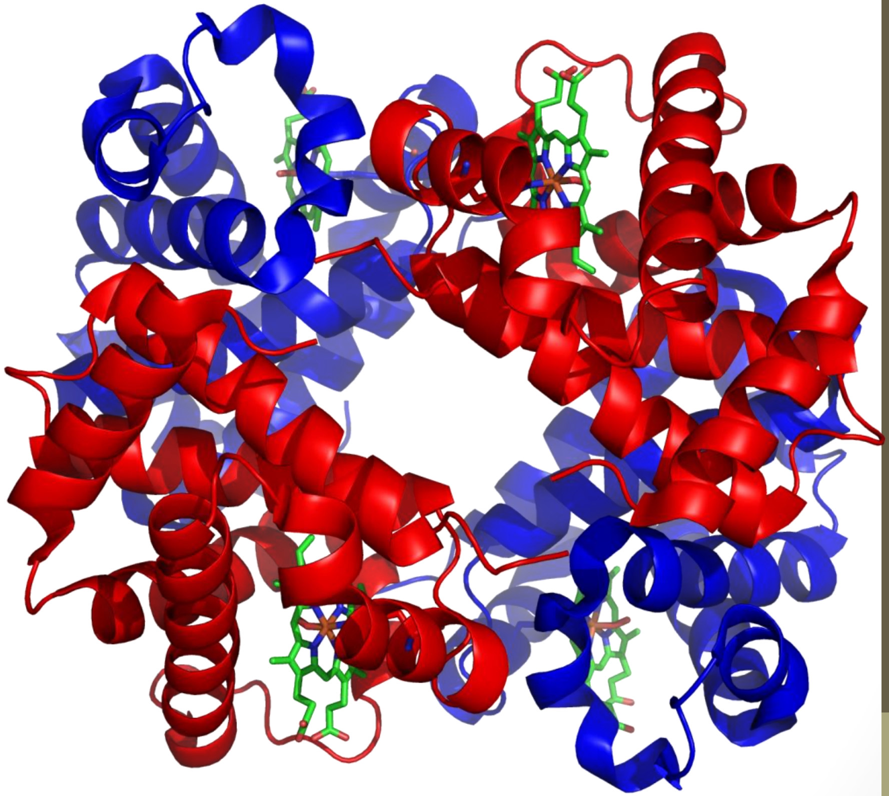
Homologs: genes (or proteins) related to another. It can be orthologue or paralogue.

Orthologs: genes (or proteins) in different species. Important in predicting function.

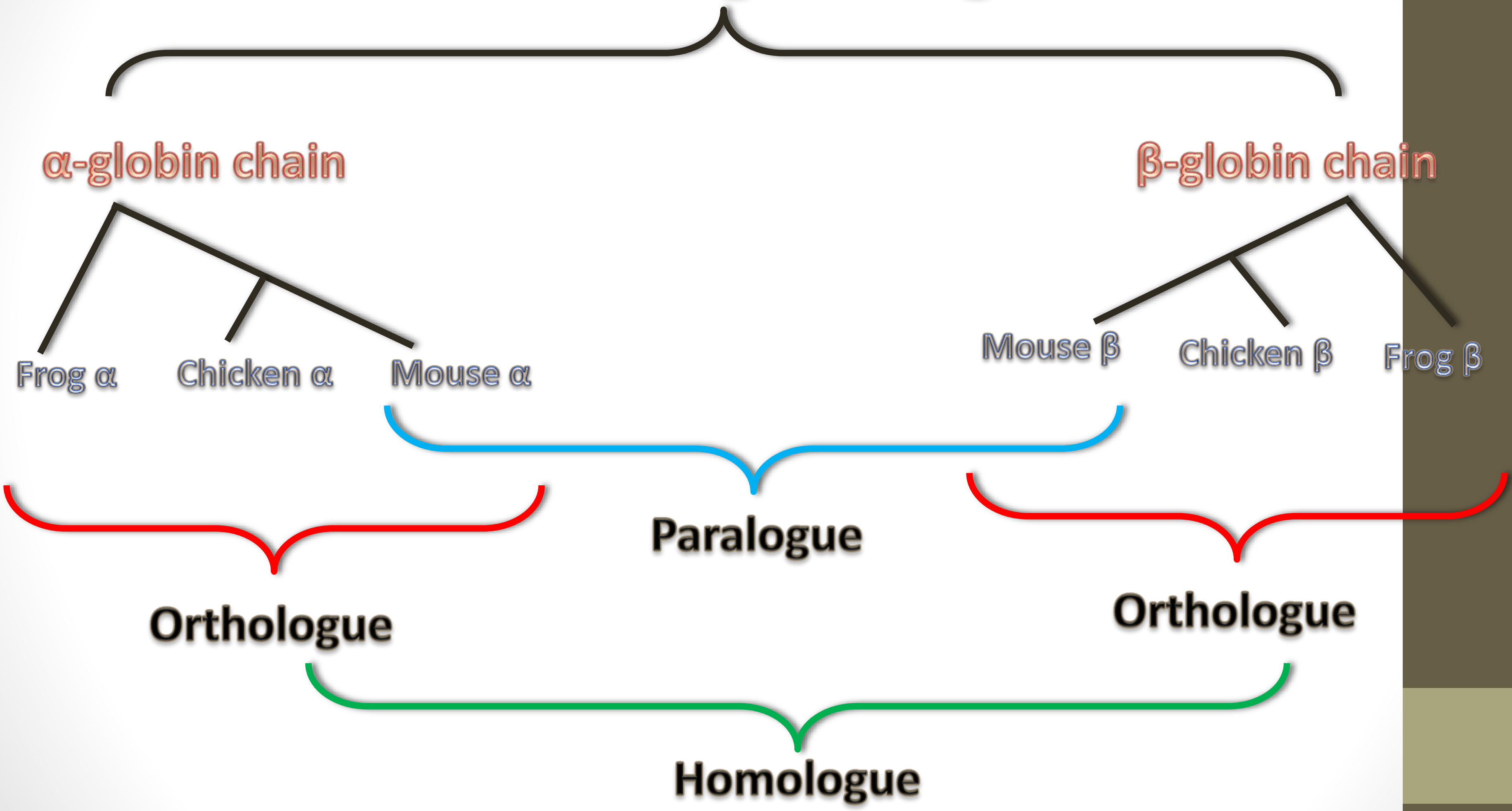
Paralogs: genes (or proteins) in the same species. They have new functions.

Example

- **Hemoglobin** has a **quaternary** structure characteristic of many **multi-subunit globular** proteins.
- It is composed ***mainly*** of:
  - **Hem (non-protein)** + protein which is 4 subunits:
  - **2 subunits** (α) and **2 subunits** (β).



Ancestral globin gene



α-globin chain

β-globin chain

Frog α

Chicken α

Mouse α

Mouse β

Chicken β

Frog β

Paralogue

Orthologue

Orthologue

Homologue

Identity

DNA/Protein sequence identity

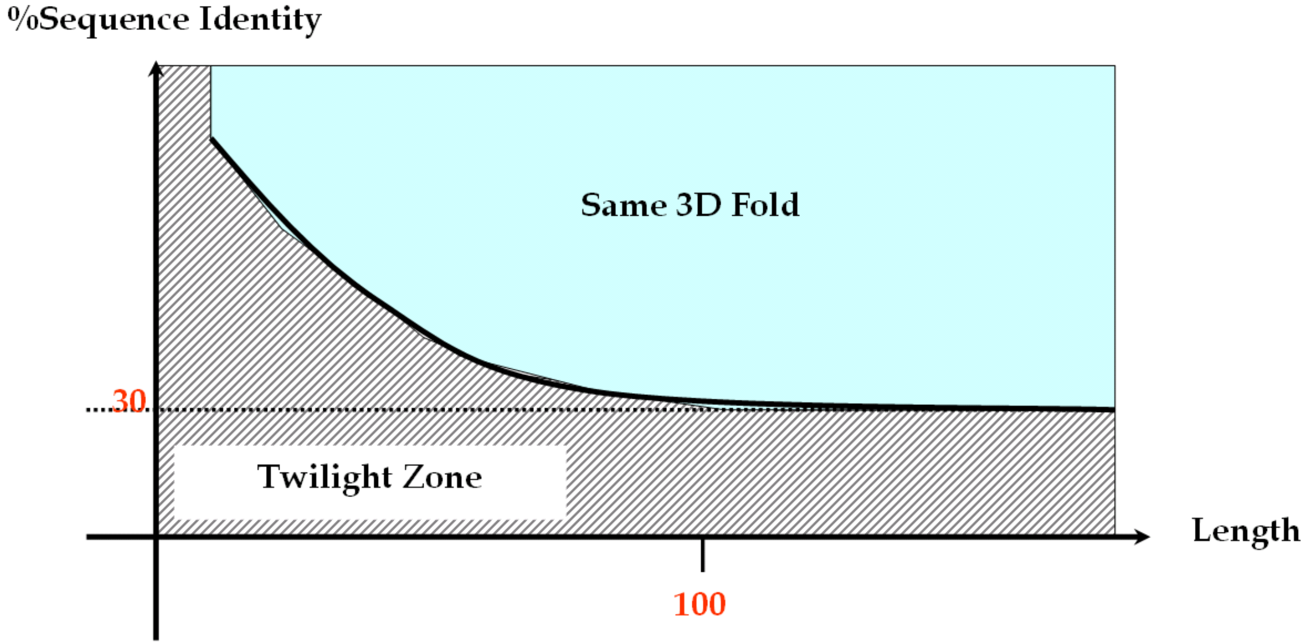
- Two protein sequences with more than 25 % identity (over 100 amino acids) are homologues

- Two DNA sequences with more than 70 % identity (over 100 nucleotides) are homologues

- Homologous sequences have
  - A common ancestor (proteins and DNA)
  - A similar 3D structure (proteins)
  - Often a similar function (proteins)

Why 25 % for proteins?

- When two proteins have less than 25% identity
  - They can be homologous or non-homologous
  - Within this range of identity, it's impossible to say which is true
- This range of identity is called the "Twilight Zone"



How to Establish Homology

- Compare your query (nucleotide or protein) with stored data in databases (such as NCBI or Uni-Prot).

- Example:
    - If the results of your search identify a Protein B to be 40% identical to your protein
    - Then, you can conclude that A and B are probably homologous if they are very similar
    - If you know the structure or the function of B, then A and B probably have the same structure

Homology, Similarity, and Identity

- Identity is a measure made on an alignment
  - **Sequence A can be "32 % identical to" Sequence B**

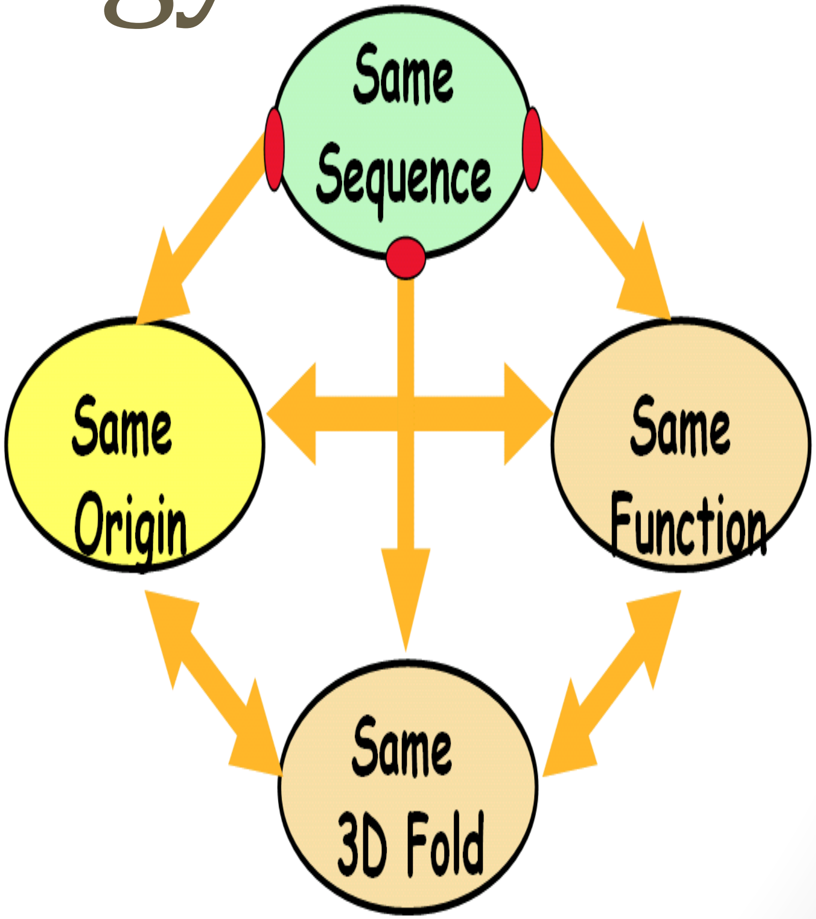
- Similarity is a measure of how close two amino acids are to identical
  - For instance, **isoleucine and leucine are similar**

- Homology is a property that exists or does not exist
  - **Sequence A IS or IS NOT homologous to Sequence B**
  - **Sequence A cannot be "40% homologous to" B**

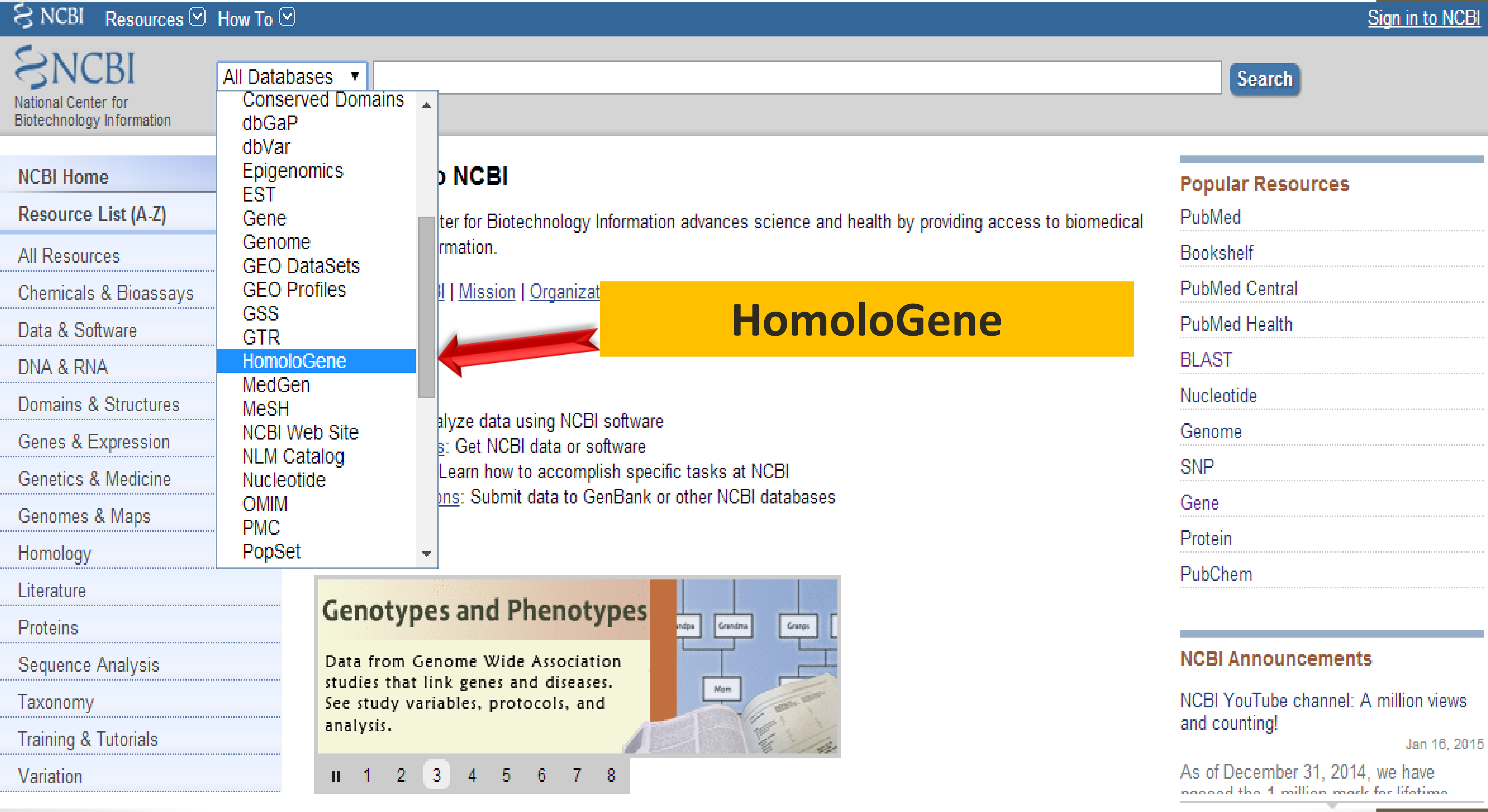
- Homology is established on the basis of measured similarity or identity

In-silico Biology

- When establishing that two proteins (A and B) are homologous, you can extrapolate everything you know from one to the other.
- It's like making a virtual experiment.
- This is in-silico biology!



HomoloGene Database



o NCBI

ter for Biotechnology Information advances science and health by providing access to biomedical rmation.

I | Mission | Organizat

alyze data using NCBI software

s: Get NCBI data or software

Learn how to accomplish specific tasks at NCBI

ons: Submit data to GenBank or other NCBI databases

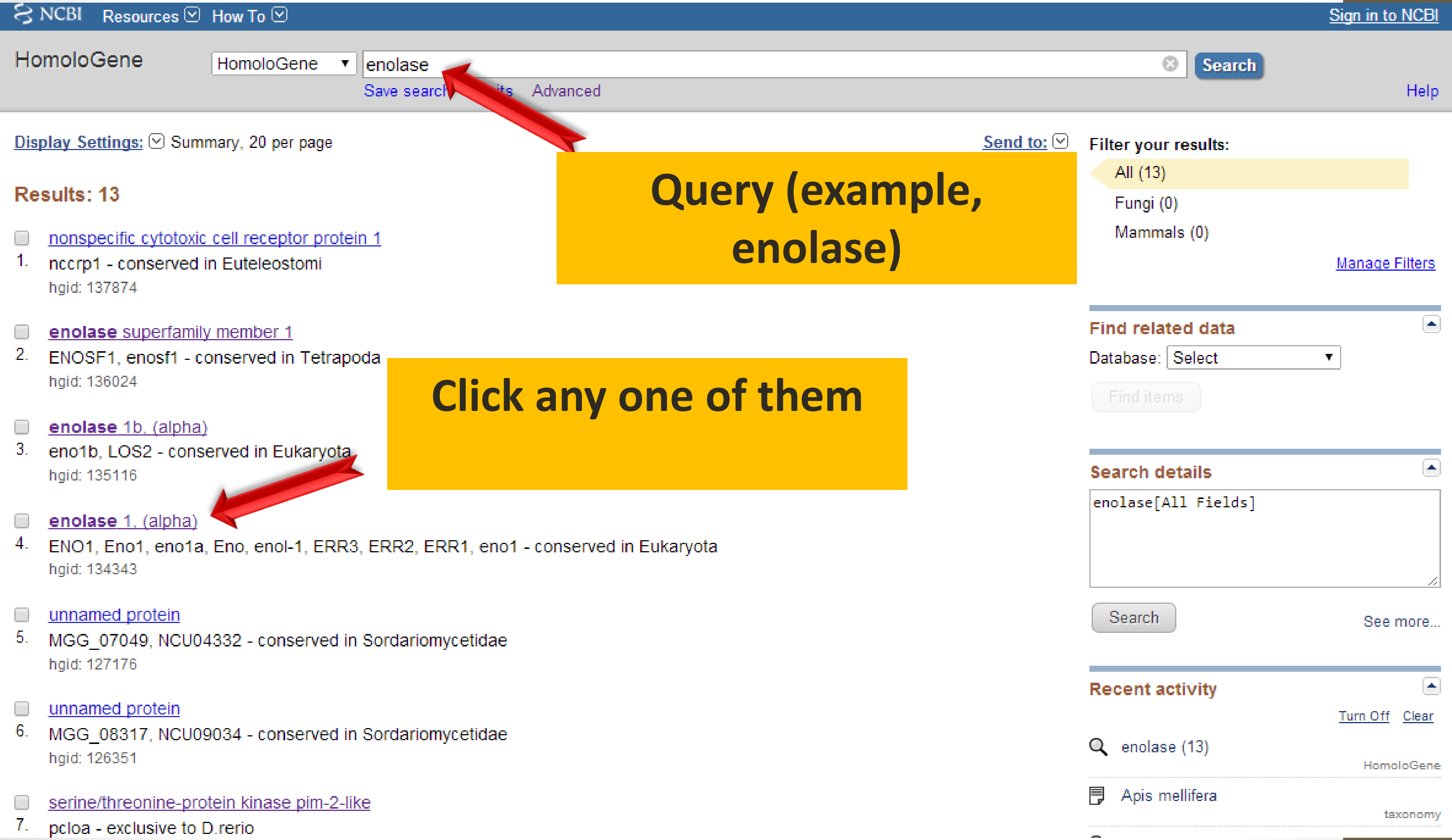
Data from Genome Wide Association studies that link genes and diseases. See study variables, protocols, and analysis.

NCBI Announcements

NCBI YouTube channel: A million views and counting!

Jan 16, 2015

As of December 31, 2014, we have passed the 1 million mark for lifetime



NCBI Resources How To Sign in to NCBI

HomoloGene

HomoloGene ▾ enolase Search

Save searc its Advanced Help

Display Settings: Summary, 20 per page Send to:

Query (example, enolase)

Results: 13

□ nonspecific cytotoxic cell receptor protein 1
1. nccrp1 - conserved in Euteleostomi
hgid: 137874

□ **enolase** superfamily member 1
2. ENOSF1, enosf1 - conserved in Tetrapoda
hgid: 136024

□ **enolase** 1b, (alpha)
3. eno1b, LOS2 - conserved in Eukaryota
hgid: 135116

Click any one of them

□ **enolase** 1, (alpha)
4. ENO1, Eno1, eno1a, Eno, enol-1, ERR3, ERR2, ERR1, eno1 - conserved in Eukaryota
hgid: 134343

□ unnamed protein
5. MGG_07049, NCU04332 - conserved in Sordariomycetidae
hgid: 127176

□ unnamed protein
6. MGG_08317, NCU09034 - conserved in Sordariomycetidae
hgid: 126351

□ serine/threonine-protein kinase pim-2-like
7. pcloa - exclusive to D.rerio

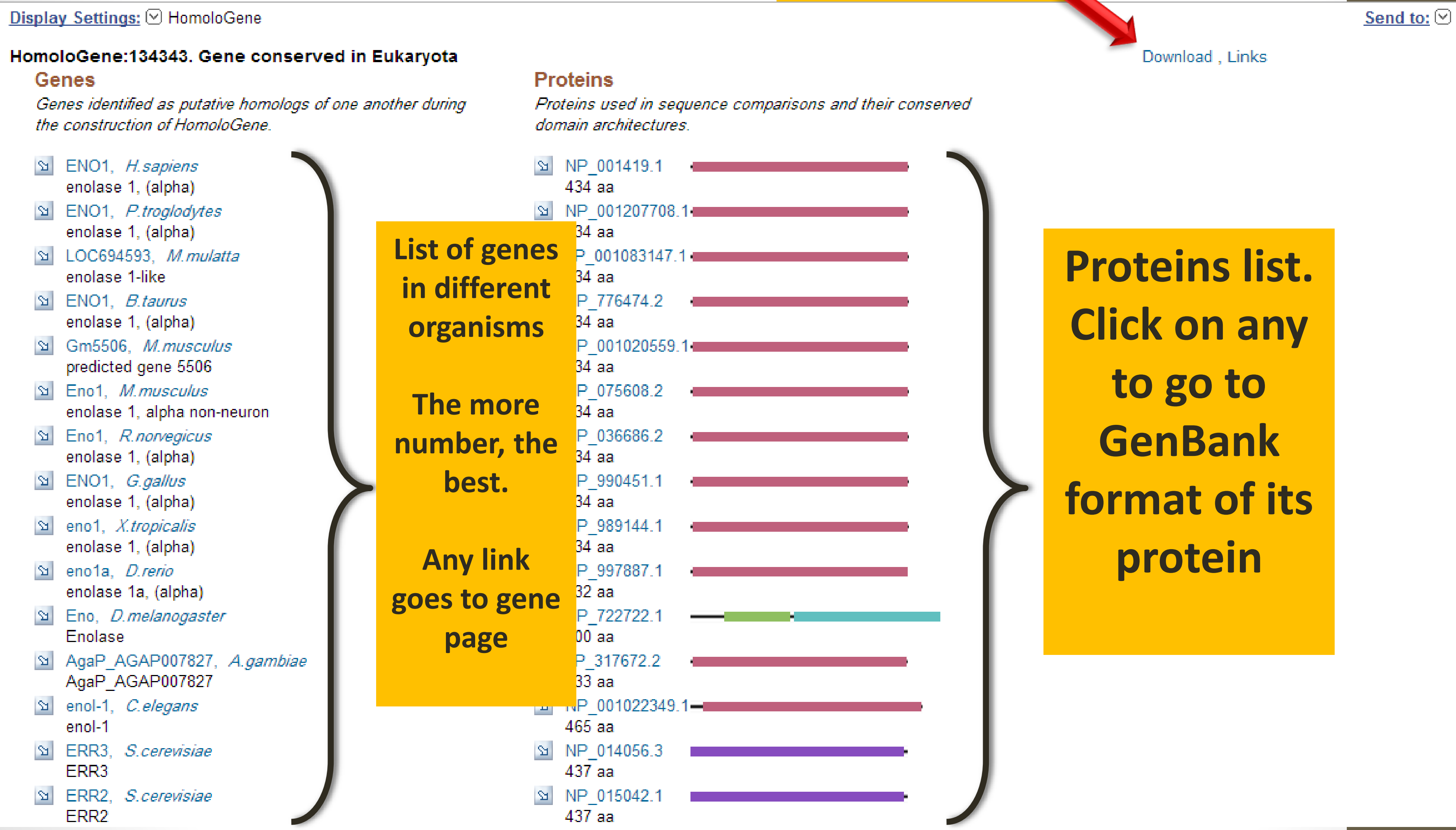
Filter your results:
All (13)
Fungi (0)
Mammals (0)
Manage Filters

Find related data
Database: Select ▾
Find items

Search details
enolase[All Fields]
Search See more...

Recent activity
Turn Off Clear
enolase (13) HomoloGene
Apis mellifera taxonomy

Download



Display Settings: HomoloGene

Send to:

Download , Links

HomoloGene:134343. Gene conserved in Eukaryota

Genes

Genes identified as putative homologs of one another during the construction of HomoloGene.

ENO1, *H.sapiens*
enolase 1, (alpha)

ENO1, *P.troglodytes*
enolase 1, (alpha)

LOC694593, *M.mulatta*
enolase 1-like

ENO1, *B.taurus*
enolase 1, (alpha)

Gm5506, *M.musculus*
predicted gene 5506

Eno1, *M.musculus*
enolase 1, alpha non-neuron

Eno1, *R.norvegicus*
enolase 1, (alpha)

ENO1, *G.gallus*
enolase 1, (alpha)

eno1, *X.tropicalis*
enolase 1, (alpha)

eno1a, *D.rerio*
enolase 1a, (alpha)

Eno, *D.melanogaster*
Enolase

AgaP_AGAP007827, *A.gambiae*
AgaP_AGAP007827

enol-1, *C.elegans*
enol-1

ERR3, *S.cerevisiae*
ERR3

ERR2, *S.cerevisiae*
ERR2

Proteins

Proteins used in sequence comparisons and their conserved domain architectures.

NP_001419.1
434 aa

NP_001207708.1
34 aa

P_001083147.1
34 aa

P_776474.2
34 aa

P_001020559.1
34 aa

P_075608.2
34 aa

P_036686.2
34 aa

P_990451.1
34 aa

P_989144.1
34 aa

P_997887.1
32 aa

P_722722.1
00 aa

P_317672.2
33 aa

NP_001022349.1
465 aa

NP_014056.3
437 aa

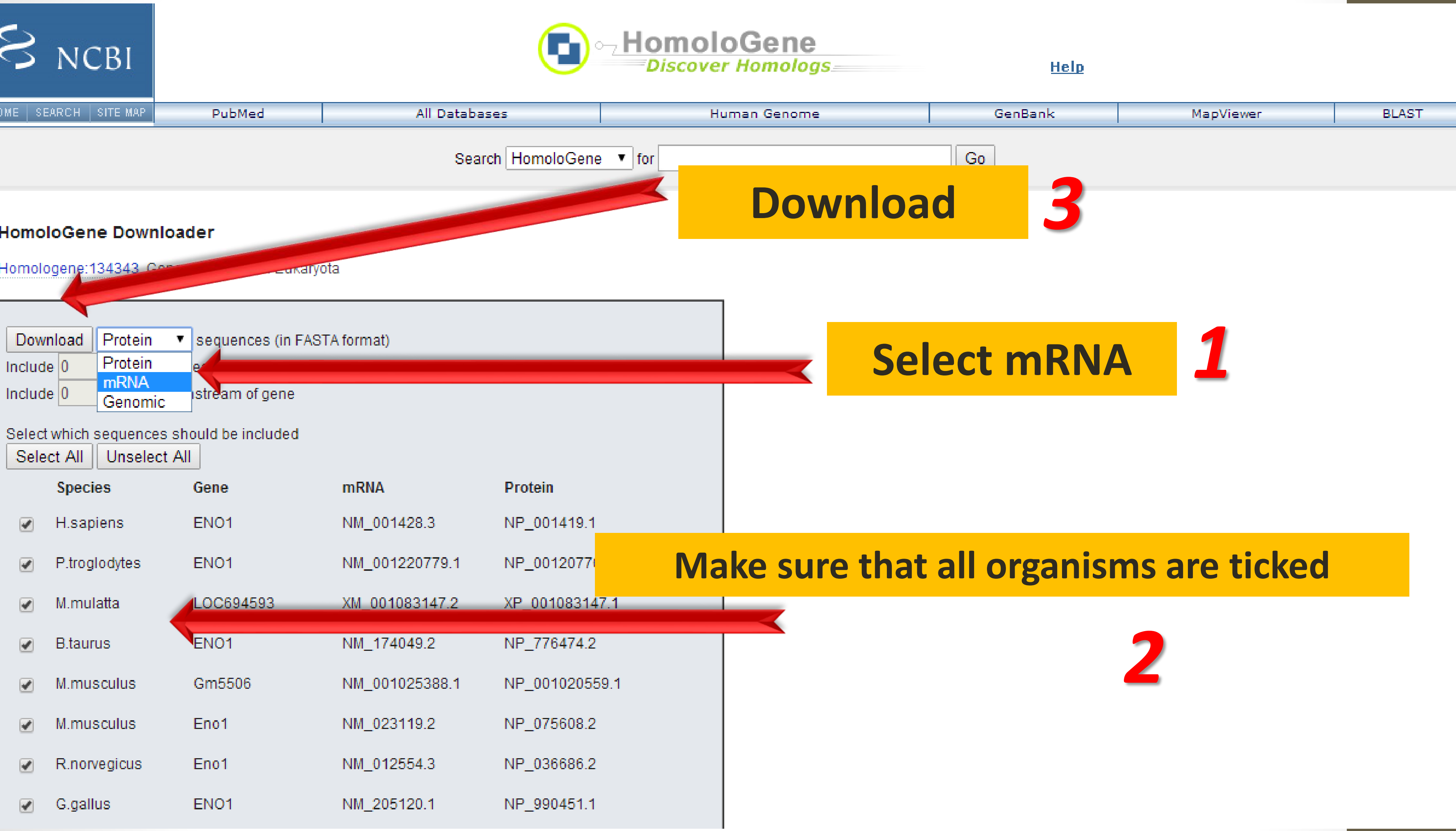
NP_015042.1
437 aa

List of genes in different organisms

The more number, the best.

Any link goes to gene page

Proteins list. Click on any to go to GenBank format of its protein



3 Download

HomoloGene Downloader

Homologene:134343. Gen Eukaryota

Download | Protein ▼ | sequences (in FASTA format)

Protein
mRNA
Genomic

Include 0 e

Include 0 stream of gene

1 Select mRNA

Select which sequences should be included

Select All | Unselect All

	Species	Gene	mRNA	Protein
☑	H.sapiens	ENO1	NM_001428.3	NP_001419.1
☑	P.troglodytes	ENO1	NM_001220779.1	NP_0012077
☑	M.mulatta	LOC694593	XM_001083147.2	XP_001083147.1
☑	B.taurus	ENO1	NM_174049.2	NP_776474.2
☑	M.musculus	Gm5506	NM_001025388.1	NP_001020559.1
☑	M.musculus	Eno1	NM_023119.2	NP_075608.2
☑	R.norvegicus	Eno1	NM_012554.3	NP_036686.2
☑	G.gallus	ENO1	NM_205120.1	NP_990451.1

2 Make sure that all organisms are ticked