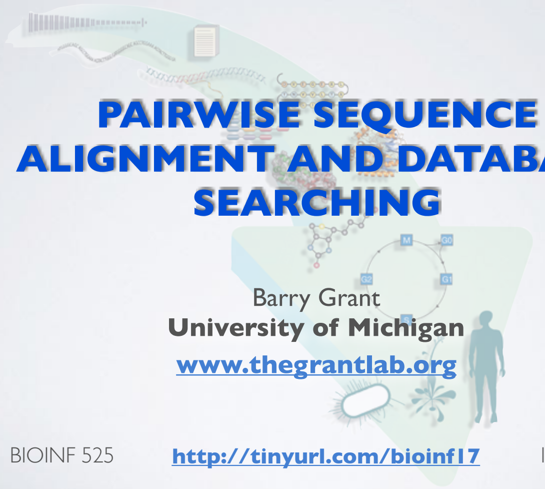


PAIRWISE SEQUENCE ALIGNMENT AND DATABASE SEARCHING

Barry Grant
University of Michigan
www.thegrantlab.org

BIOINF 525 http://tinyurl.com/bioinf17 17-Jan-2017

MODULE OVERVIEW

Objective: Provide an introduction to the practice of bioinformatics as well as a practical guide to using common bioinformatics databases and algorithms

- 1.1. *Introduction to Bioinformatics*
- 1.2. *Sequence Alignment and Database Searching*
- 1.3 *Structural Bioinformatics*
- 1.4 *Genome Informatics: High Throughput Sequencing Applications and Analytical Methods*

WEEK ONE REVIEW

☑ **Answers to last weeks homework** (19/20):
[Answers week 1]()

☑ **Muddy Point Assessment** (14/20):
[Responses]()

- Need for FASTA header lines “>example1”

- More on protein structure viewing and NGL…

- “what does the AU assembly mean?

- “Great first lab!” … Nice Assignment”.

THIS WEEK’S HOMEWORK

☑ Check out the “**Background Reading**” material online:
[Dynamic Programming]()
[Database Searching]()

☑ Complete the **lecture 1.2 homework questions**:
http://tinyurl.com/bioinf525-quiz2

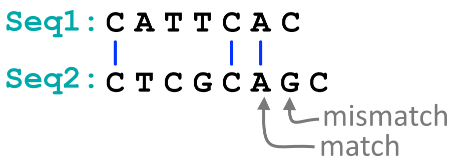
TODAYS MENU

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing
  (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - PSI-BLAST and HMM approaches

Basic Idea: Display one sequence above another with spaces (termed **gaps**) inserted in both to reveal **similarity** of nucleotides or amino acids.

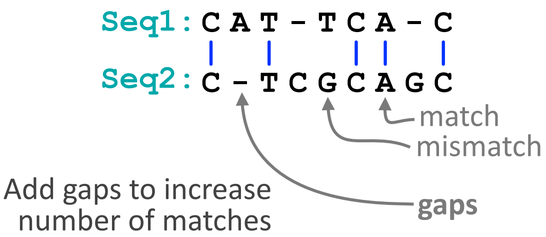
```
Seq1: C A T T C A C
Seq2: C T C G C A G C
```

Basic Idea: Display one sequence above another with spaces (termed **gaps**) inserted in both to reveal **similarity** of nucleotides or amino acids.



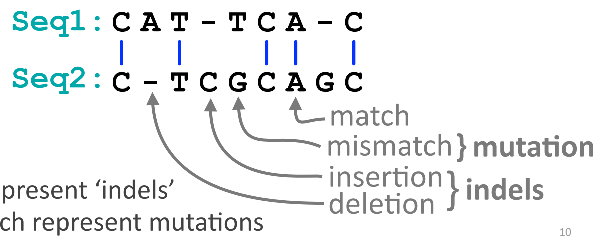
Two types of character correspondence

Basic Idea: Display one sequence above another with spaces (termed **gaps**) inserted in both to reveal **similarity** of nucleotides or amino acids.



Add gaps to increase number of matches

Basic Idea: Display one sequence above another with spaces (termed **gaps**) inserted in both to reveal **similarity** of nucleotides or amino acids.



Gaps represent ‘indels’
mismatch represent mutations

Why compare biological sequences?

- To obtain **functional or mechanistic insight** about a sequence by inference from another potentially better characterized sequence
- To find whether two (or more) genes or proteins are **evolutionarily related**
- To find **structurally or functionally similar regions** within sequences (e.g. catalytic sites, binding sites for other molecules, etc.)
- Many practical bioinformatics applications…

Practical applications of sequence alignment include...

- **Similarity searching of databases**
  - Protein structure prediction, annotation, etc...
- **Assembly of sequence reads** into a longer construct such as a genomic sequence
- **Mapping sequencing reads to a known genome**
  - "Resequencing", looking for differences from reference genome - SNPs, indels (insertions or deletions)
  - Mapping transcription factor binding sites via ChIP-Seq (chromatin immuno-precipitation sequencing)
  - Pretty much all next-gen sequencing data analysis

Practical applications of sequence alignment include...

- **Similarity searching of databases**
  - Protein structure predictio
- **Assembly of sequen** construct such as a bac
- **Map s to a known genome**
  - ooking for differences from reference NPs, indels (insertions or deletions)
  - apping transcription factor binding sites via ChIP-Seq (chromatin immuno-precipitation sequencing)
  - Pretty much all next-gen sequencing data analysis

N.B. Pairwise sequence alignment is arguably the most fundamental operation of bioinformatics!

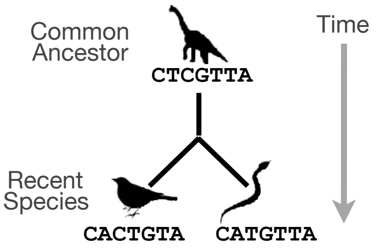
Outline for today

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - PSI-BLAST and HMM approaches

Sequence comparison is most informative when it detects **homologs**

Homologs are sequences that have common origins *i.e.* they share a **common ancestor**

- They may or may not have common activity



Can be used to establish evolutionary relationships

Key terms

When we talk about related sequences we use specific terminology.

Homologous sequences may be either:

– **Orthologs** or **Paralogs**

(Note. these are all or nothing relationships!)

Any pair of sequences may share a certain level of:

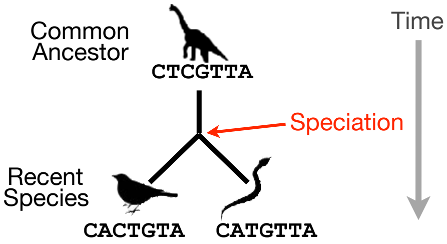
– **Identity** and/or **Similarity**

(Note. if these metrics are above a certain level we often infer homology)

Orthologs tend to have similar function

Orthologs: are homologs produced by speciation that have diverged due to divergence of the organisms they are associated with.

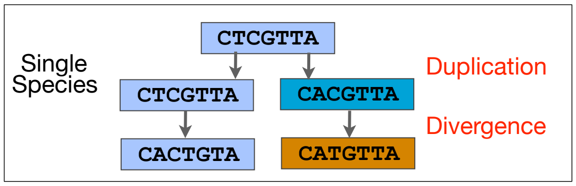
– Ortho = [greek: straight] ... implies direct descent



Paralogs tend to have slightly different functions

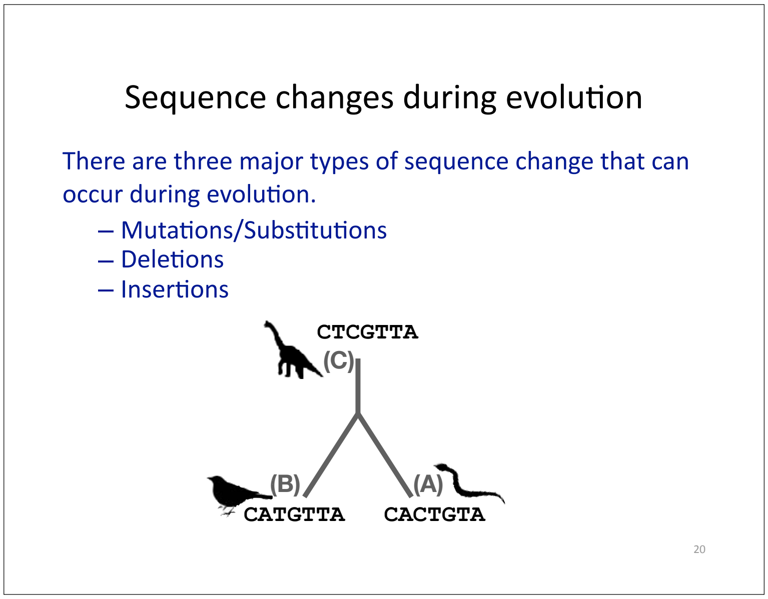
Paralogs: are homologs produced by **gene duplication**. They represent genes derived from a common ancestral gene that *duplicated within an organism* and then subsequently *diverged by accumulated mutation*.

– Para = [greek: along side of]



Orthologs *vs* Paralogs

- In practice, determining ortholog *vs* paralog can be a complex problem:
  – gene loss after duplication,
  – lack of knowledge of evolutionary history,
  – weak similarity because of evolutionary distance
- Homology does not necessarily imply exact same function
  – may have similar function at very crude level but play a different physiological role



Sequence changes during evolution

There are three major types of sequence change that can occur during evolution.

- Mutations/Substitutions
- Deletions
- Insertions

CTCGTTA

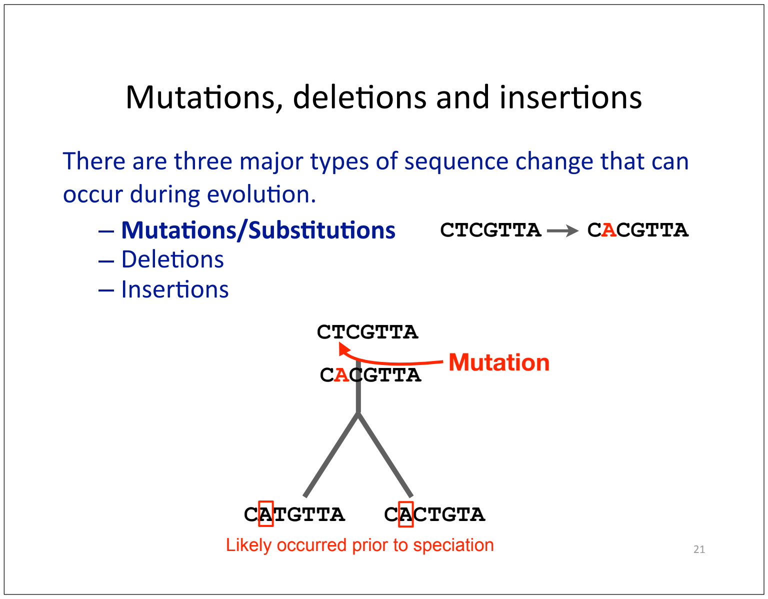
(C)

(B)

(A)

CATGTTA

CACTGTA



Mutations, deletions and insertions

There are three major types of sequence change that can occur during evolution.

- **Mutations/Substitutions** CTCGTTA → CACGTTA
- Deletions
- Insertions

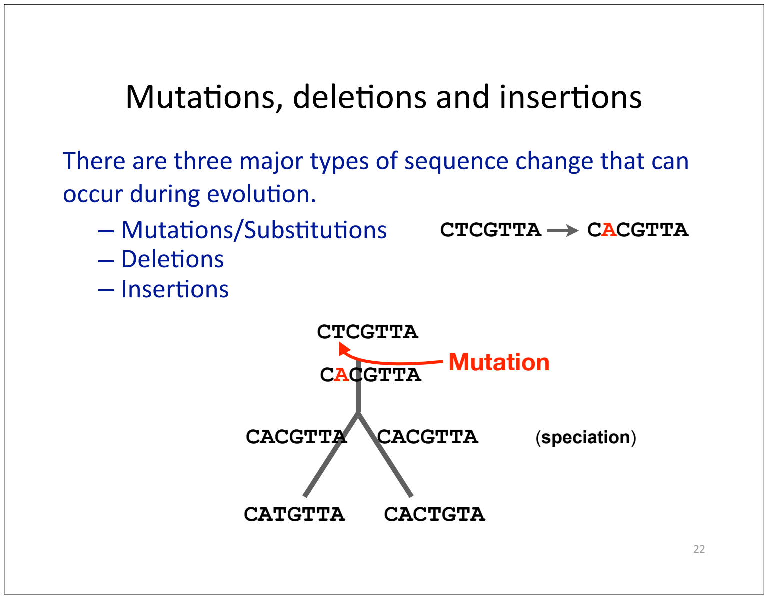
CTCGTTA

Mutation

CACGTTA

CATGTTA CACTGTA

Likely occurred prior to speciation



Mutations, deletions and insertions

There are three major types of sequence change that can occur during evolution.

- Mutations/Substitutions CTCGTTA → CACGTTA
- Deletions
- Insertions

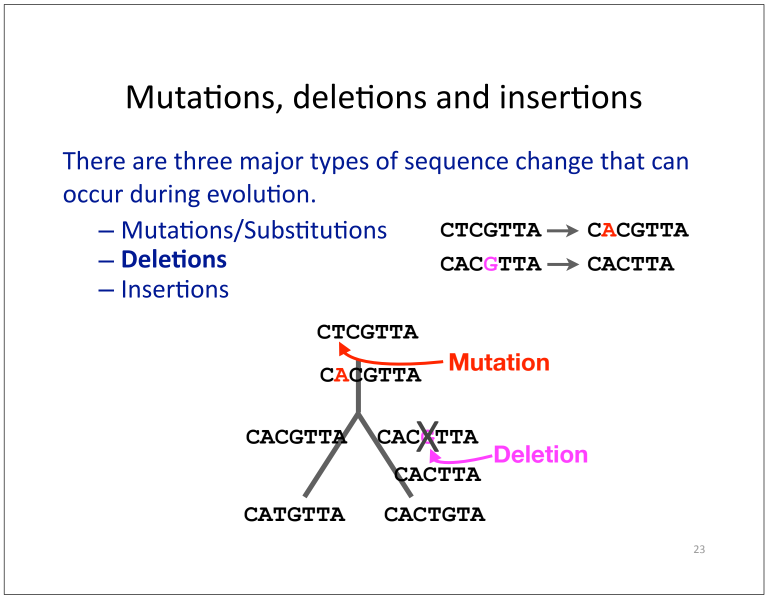
CTCGTTA

Mutation

CACGTTA

CACGTTA CACGTTA (speciation)

CATGTTA CACTGTA



Mutations, deletions and insertions

There are three major types of sequence change that can occur during evolution.

- Mutations/Substitutions CTCGTTA → CACGTTA
- **Deletions** CACGTTA → CACTTA
- Insertions

CTCGTTA

Mutation

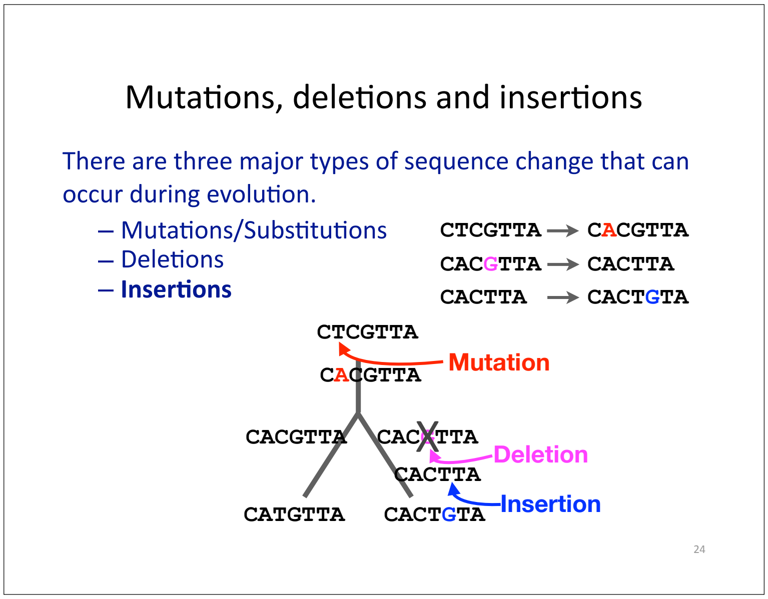
CACGTTA

CACGTTA CAC~~G~~TTA

Deletion

CACTTA

CATGTTA CACTGTA



Mutations, deletions and insertions

There are three major types of sequence change that can occur during evolution.

- Mutations/Substitutions CTCGTTA → CACGTTA
- Deletions CACGTTA → CACTTA
- **Insertions** CACTTA → CACTGTA

CTCGTTA

Mutation

CACGTTA

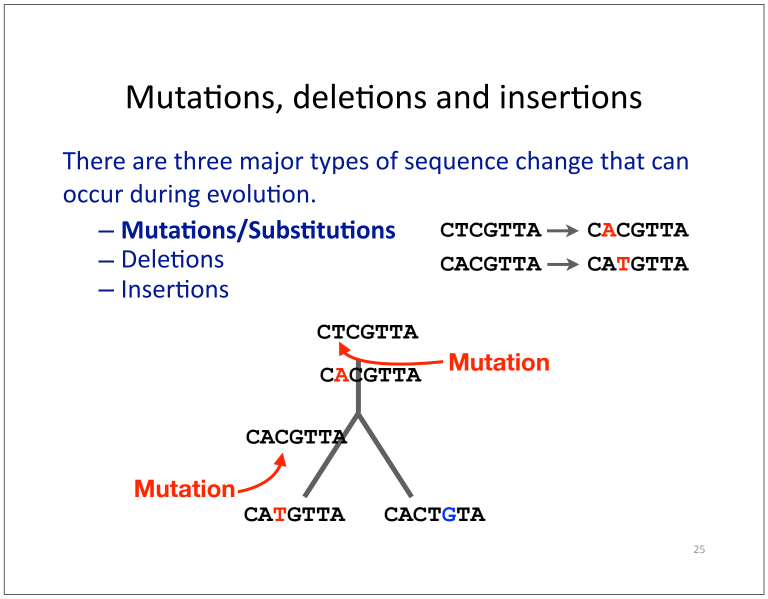
CACGTTA CAC~~G~~TTA

Deletion

CACTTA

Insertion

CATGTTA CACTGTA



Mutations, deletions and insertions

There are three major types of sequence change that can occur during evolution.

- **Mutations/Substitutions** CTCGTTA → CACGTTA
- Deletions CACGTTA → CATGTTA
- Insertions

CTCGTTA

Mutation

CACGTTA

CACGTTA

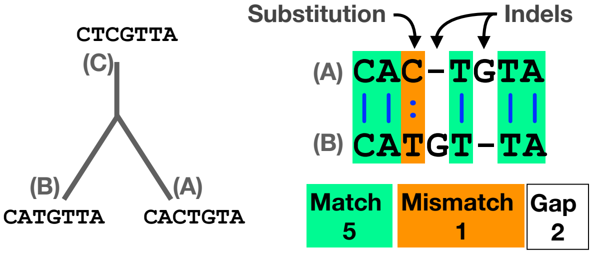
Mutation

CATGTTA CACTGTA

Alignment view

Alignments are great tools to visualize sequence similarity and evolutionary changes in homologous sequences.

- **Mismatches** represent mutations/substitutions
- **Gaps** represent insertions and deletions (indels)



Alternative alignments

- Unfortunately, finding the correct alignment is difficult if we do not know the evolutionary history of the two sequences
  - There are many possible alignments
  - Which alignment is best?

```
CACTGTA    CACTGT-A    CAC-TGTA
||:::||    || ||| |    ||: | ||
CATGTTA    CA-TGTTA    CATGT-TA
```

Alternative alignments

- One way to judge alignments is to compare their number of matches, insertions, deletions and mutations

🟩 4 matches 🟧 3 mismatches ○ 0 gaps	🟩 6 matches 🟧 0 mismatches ○ 2 gaps	🟩 5 matches 🟧 1 mismatches ○ 2 gaps

```
CACTGTA    CACTGT-A    CAC-TGTA
||:::||    || ||| |    ||: | ||
CATGTTA    CA-TGTTA    CATGT-TA
```

Scoring alignments

- We can assign a score for each match (+3), mismatch (+1) and indel (-1) to identify the **optimal alignment** *for this scoring scheme*

🟩 4 (+3) 🟧 3 (+1) ○ 0 (-1) = 15	🟩 6 (+3) 🟧 0 (+1) ○ 2 (-1) = 16	🟩 5 (+3) 🟧 1 (+1) ○ 2 (-1) = 14

```
CACTGTA    CACTGT-A    CAC-TGTA
||:::||    || ||| |    ||: | ||
CATGTTA    CA-TGTTA    CATGT-TA
```

Optimal alignments

- Biologists often prefer **parsimonious alignments**, where the number of postulated sequence changes is minimized.

🟩 4 matches 🟧 3 mismatches ○ 0 gaps	🟩 6 matches 🟧 0 mismatches ○ 2 gaps	🟩 5 matches 🟧 1 mismatches ○ 2 gaps

```
CACTGTA    CACTGT-A    CAC-TGTA
||:::||    || ||| |    ||: | ||
CATGTTA    CA-TGTTA    CATGT-TA
```

Optimal alignments

- Biologists often prefer **parsimoniou** **alignments**, where the numb sequence changes is

Warning: There may be more than one optimal alignment and these may not reflect the true evolutionary history of our sequences!

🟩 4 matches 🟧 3 mi		matches 🟧 1 mismatches ○ 2 gaps

```
           CACTGT-A    CAC-TGTA
           || ||| |    ||: | ||
 ATGTTA    CA-TGTTA    CATGT-TA
```

Side note: sequence *identity* and *similarity*

- Two commonly quoted metrics for pairs of aligned sequences.
  - **Sequence identity**: typically quotes the percent of identical characters in the aligned region of two sequences
  - **Sequence similarity**: typically the score resulting from optimal pair-wise alignment (note dependence on parameters used: *i.e.* scoring scheme)
- N.B. In contrast, **homology is an all or nothing relationship**, you can not have a percent homology!

Side note: sequence identity and similarity

- High sequence similarity is frequently used as an indicator of homology
  - Use to find genes and/or proteins with potentially similar or identical function
  - Can query a database of sequences by performing a series of pair-wise alignments
- Knowledge of the difference between sequences can also yield valuable functional and mechanistic insights
  - A gene from a normal and an affected subject – possible cause of a heritable disease
  - Similar proteins with different substrate specificities – what amino acid changes might be responsible for this?

Outline for today

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - PSI-BLAST and HMM approaches

Outline for today

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods

How do we compute the optimal alignment between two sequences?

(global vs local alignment)

- Rapid heu
  - BLAST
- Practical d
  - PSI-BL

Quiz questions:

http://tinyurl.com/bioinf525-quiz2

Pair-wise Sequence Alignment

- **Objective**: arrange two sequences in such a fashion that pairs of matching characters between the two sequences are maximized
  - Match does not have to be identity, can be defined by a function that ranks or scores the characters being compared (often termed a **substitution matrix**)
  - Ungapped alignment example – bars indicate matching characters

```
Seq1: GTAATCTG-
       |||  |||
Seq2: -TAAGCTGA
```

Simplest case – brute force alignments

- In the simplest case we can simply slide one sequence across the other and count matching characters for each possible alignment
  - Chose a scoring scheme and do not allow internal gaps within sequences
  - Algorithmic complexity is linear

    N + M alignments to consider

    (where N and M are the length of each sequence)

Brute Force Alignment, No Gaps

```
GTAATCTG
       TTAAGCTGA
```

```
 GTAATCTG
  | |
 TTAAGCTGA
```

```
GTAATCTG
      |
      TTAAGCTGA
```

```
GTAATCTG
 ||| |||
TTAAGCTGA
```

```
GTAATCTG
      |
     TTAAGCTGA
```

```
 GTAATCTG
  ||
TTAAGCTGA
```

```
GTAATCTG
    TTAAGCTGA
```

```
  GTAATCTG
      |
TTAAGCTGA
```

```
GTAATCTG
    |  |
   TTAAGCTGA
```

```
   GTAATCTG
TTAAGCTGA
```

```
GTAATCTG
  TTAAGCTGA
```

Etc...

Slide from Jeffery de Wet

Gaps make the brute force method unusable for all but the shortest sequences

- Pairs of related sequences often have insertions or deletions relative to one-another, we therefore require **gapped pair-wise alignment**
  - Need to generate all the possible gap lengths and combinations of gaps at all possible positions in both sequences
  - For two sequences of equal length, the formula is:

$$\binom{2N}{N} = \frac{(2N)!}{(N!)^2} \simeq \frac{2^{2N}}{\sqrt{\pi N}}$$

N = 10: 184756
N = 50: ~1.00E29
N = 250: ~1.17E149

Slide from Jeffery de Wet

Three general solutions to the alignment problem

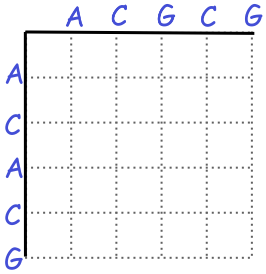
- The **dot plot** or **dot matrix** approach
  - A simple graphical method for pair-wise alignment
  - No scoring, so difficult to compare alternative alignments
  - Can give visual clues to sequence structure but requires human interaction
- **Dynamic programming** algorithms
  - Provides Optimal solutions (but not necessarily unique solutions)
- Heuristic **word** or **k-tuple** approaches
  - Much faster (e.g. **BLAST** and **FASTA**)
  - Widely used for database searches
  - May miss some pairs with low similarity

Three general solutions to the alignment problem

- The **dot plot** or **dot matrix** approach
  - A simple graphical method for pair-wise alignment
  - No scoring, so difficult to compare alternative alignments
  - Can give visual clues to sequence structure but requires human interaction
- **Dynamic programming** algorithms
  - Provides Optimal solutions (but not necessarily unique solutions)
- Heuristic **word** or **k-tuple** approaches
  - Much faster (e.g. **BLAST** and **FASTA**)
  - Widely used for database searches
  - May miss some pairs with low similarity

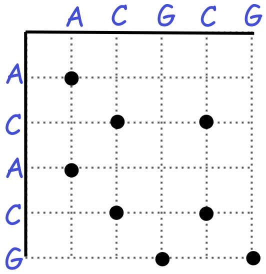
Dot plots: simple graphical approach

- Place one sequence on the vertical axis of a 2D grid (or matrix) and the other on the horizontal



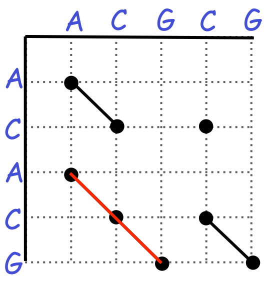
Dot plots: simple graphical approach

- Now simply put dots where the horizontal and vertical sequence values match



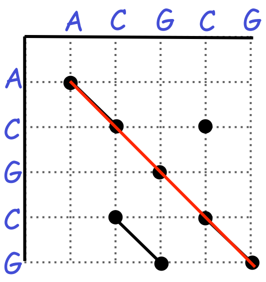
Dot plots: simple graphical approach

- Diagonal runs of dots indicate matched segments of sequence



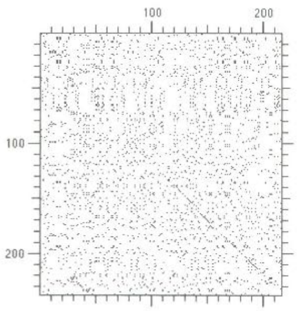
Dot plots: simple graphical approach

Q. What would the dot matrix of a two identical sequences look like?



Dot plots: simple graphical approach

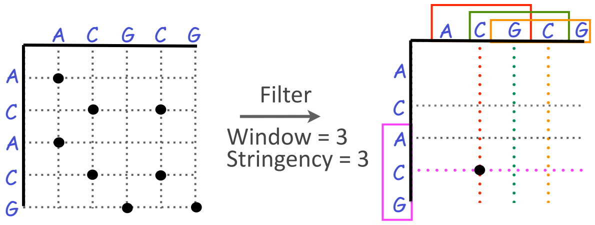
- Dot matrices for long sequences can be noisy



Dot plots: window size and match stringency

Solution: use a window and a threshold

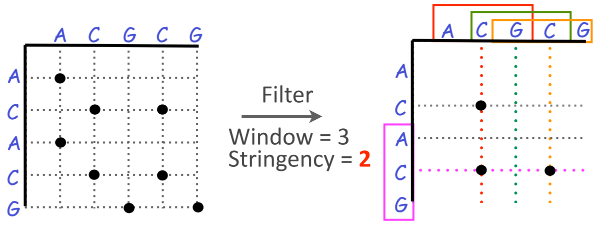
- compare character by character within a window
- require certain fraction of matches within window in order to display it with a dot.
  - You have to choose window size and stringency



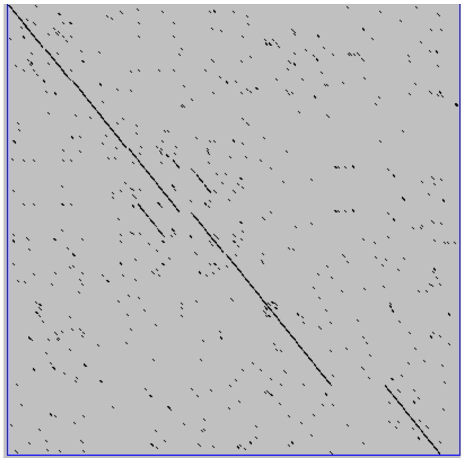
Dot plots: window size and match stringency

Solution: use a window and a threshold

- compare character by character within a window
- require certain fraction of matches within window in order to display it with a dot.
  - You have to choose window size and stringency



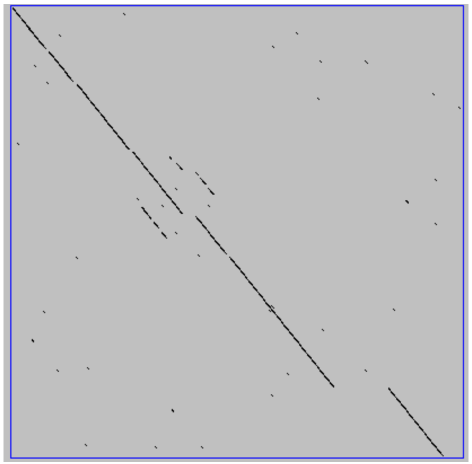
Window size = 5 bases



A dot plot simply puts a dot where two sequences match. In this example, dots are placed in the plot if 5 bases in a row match perfectly. Requiring a 5 base perfect match is a **heuristic** – only look at regions that have a certain degree of identity.

Do you expect evolutionarily related sequences to have more word matches (matches in a row over a certain length) than random or unrelated sequences?

Window size = 7 bases



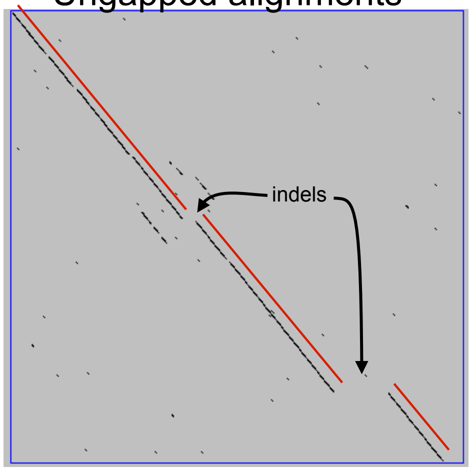
This is a dot plot of the same sequence pair. Now 7 bases in a row must match for a dot to be place. Noise is reduced.

Using windows of a certain length is very similar to using words (kmers) of N characters in the heuristic alignment search tools

Bigger window (kmer) fewer matches to consider

Web site used: http://www.vivo.colostate.edu/molkit/dnadot/

Ungapped alignments

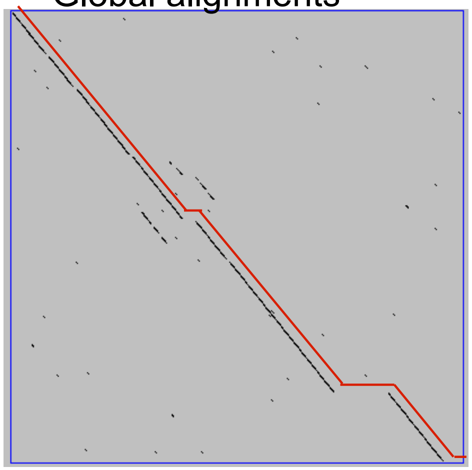


Only **diagonals** can be followed.

Downward or rightward paths represent **insertion** or **deletions** (gaps in one sequence or the other).

Web site used: http://www.vivo.colostate.edu/molkit/dnadot/

Global alignments



Global alignments go from end to end, *i.e.* from the upper left corner to the lower right corner.

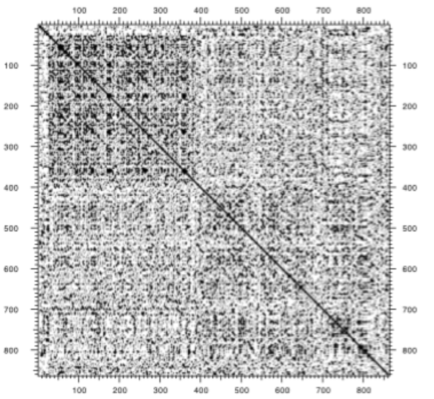
Global alignments do not have good statistical characterization and are **not used for database searches.**

Web site used: http://www.vivo.colostate.edu/molkit/dnadot/

Uses for dot matrices

- Visually assessing the similarity of two protein or two nucleic acid sequences
- Finding local repeat sequences within a larger sequence by comparing a sequence to itself
  - Repeats appear as a set of diagonal runs stacked vertically and/or horizontally

Repeats

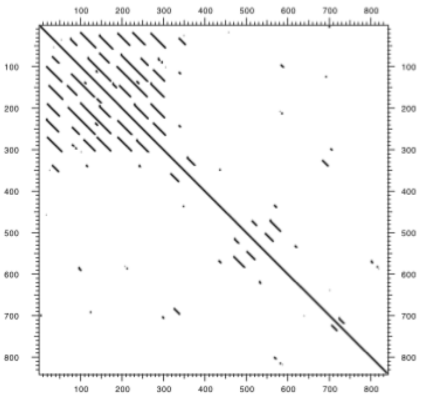


Human LDL receptor protein sequence (Genbank P01130)

W = 1
S = 1

(Figure from Mount, “Bioinformatics sequence and genome analysis”)

Repeats



Human LDL receptor protein sequence (Genbank P01130)

W = 23
S = 7

(Figure from Mount, “Bioinformatics sequence and genome analysis”)

Side note: dots can have "weights"

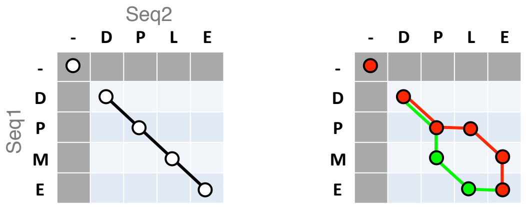
- Some matches can be rewarded more than others, depending on likelihood
- Use PAM or BLOSUM **substitution matrix**
  - (more on these later)
- Put a dot only if a minimum total or average weight is achieved
  - See chapter 3 in *Mount, "Bioinformatics sequence and genome analysis"*.

Three general solutions to the alignment problem

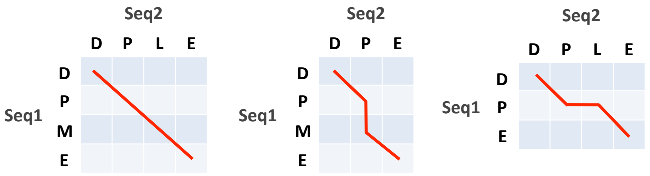
- The **dot plot** or **dot matrix** approach
  - A simple graphical method for pair-wise alignment
  - No scoring, so difficult to compare alternative alignments
  - Can give visual clues to sequence structure but requires human interaction
- **Dynamic programming** algorithms
  - Provides Optimal solutions (but not necessarily unique solutions)
- Heuristic **word** or **k-tuple** approaches
  - Much faster (e.g. **BLAST** and **FASTA**)
  - Widely used for database searches
  - May miss some pairs with low similarity

The Dynamic Programming Algorithm

- The dynamic programming algorithm can be thought of an extension to the dot plot approach
  - One sequence is placed down the side of a grid and another across the top
  - Instead of placing a dot in the grid, we **compute a score** for each position
  - Finding the optimal alignment corresponds to finding the path through the grid with the **highest possible score**



Different paths represent different alignments

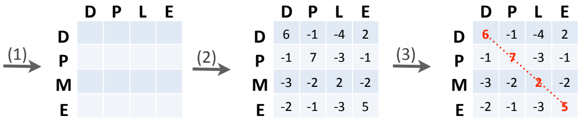


```
Seq1: D P L E     Seq1: D P M E     Seq1: D P – E
      | | : |           | |   |           | |   |
Seq2: D P M E     Seq2: D P – E     Seq2: D P L E
```

Matches are represented by diagonal paths and indels with horizontal or vertical path segments

Algorithm of **Needleman and Wunsch**

- The Needleman–Wunsch approach to global sequence alignment has three basic steps:
  - (1) setting up a 2D-grid (or **alignment matrix**),
  - (2) **scoring the matrix**, and
  - (3) identifying the **optimal path** through the matrix



	D	P	L	E
D				
P				
M				
E				

	D	P	L	E
D	6	-1	-4	2
P	-1	7	-3	-1
M	-3	-2	2	-2
E	-2	-1	-3	5

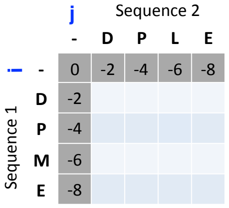
	D	P	L	E
D	6	-1	-4	2
P	-1	7	-3	-1
M	-3	-2	2	-2
E	-2	-1	-3	5

Needleman, S.B. & Wunsch, C.D. (1970) "A general method applicable to the search for similarities in the amino acid sequences of two proteins." J. Mol. Biol. 48:443-453.

Scoring the alignment matrix

- Start by filling in the first row and column – these are all indels (gaps).
  - Each step you take you will add the **gap penalty** to the score ($S_{i,j}$) accumulated in the previous cell

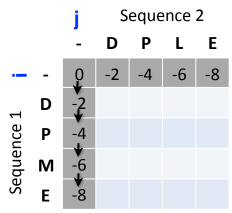
<u>**Scores**</u>: match = +1, mismatch = -1, gap = -2


j
i
Sequence 2
Sequence 1

	-	D	P	L	E
-	0	-2	-4	-6	-8
D	-2				
P	-4				
M	-6				
E	-8				

Scoring the alignment matrix

- Start by filling in the first row and column – these are all indels (gaps).
  - Each step you take you will add the **gap penalty** to the score ($S_{i,j}$) accumulated in the previous cell



j Sequence 2

Sequence 1

		-	D	P	L	E
i	-	0	-2	-4	-6	-8
	D	-2				
	P	-4				
	M	-6				
	E	-8				

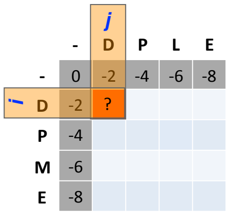
Scores: match = +1, mismatch = -1, gap = -2

$$S_{i+4} = (-2) + (-2) + (-2) + (-2)$$

```
Seq1: DPME
Seq2: ----
```

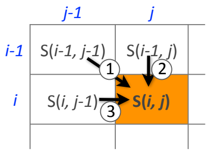
Scoring the alignment matrix

- Then go to the empty corner cell (upper left). It has filled in values in up, left and diagonal directions
  - Now can ask which of the three directions gives the highest score?
  - keep track of this score and direction



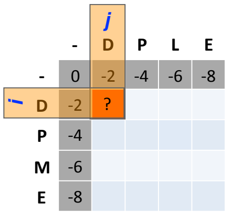
			j			
		-	D	P	L	E
	-	0	-2	-4	-6	-8
i	D	-2	?			
	P	-4				
	M	-6				
	E	-8				

Scores: match = +1, mismatch = -1, gap = -2



Scoring the alignment matrix

- Then go to the empty corner cell (upper left). It has filled in values in up, left and diagonal directions
  - Now can ask which of the three directions gives the highest score?
  - keep track of this score and direction



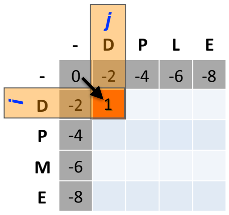
			j			
		-	D	P	L	E
	-	0	-2	-4	-6	-8
i	D	-2	?			
	P	-4				
	M	-6				
	E	-8				

Scores: match = +1, mismatch = -1, gap = -2

$$S(i, j) = \text{Max}\begin{cases} S(i\text{-}1, j\text{-}1) + \text{(mis)match} & \searrow \; \textcircled{1} \\ S(i\text{-}1, j) - \text{gap penalty} & \downarrow \; \textcircled{2} \\ S(i, j\text{-}1) - \text{gap penalty} & \rightarrow \; \textcircled{3} \end{cases}$$

Scoring the alignment matrix

- Then go to the empty corner cell (upper left). It has filled in values in up, left and diagonal directions
  - Now can ask which direction gives the highest score
  - keep track of direction and score



			j			
		-	D	P	L	E
	-	0	-2	-4	-6	-8
i	D	-2	1			
	P	-4				
	M	-6				
	E	-8				

Scores: match = +1, mismatch = -1, gap = -2

↘ ① (0)+(+1) = +1 <= (D-D) match!

↓ ② (-2)+(-2) = -4

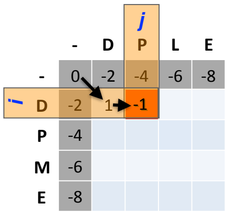
→ ③ (-2)+(-2) = -4

Alignment

```
D
D
```

Scoring the alignment matrix

- At each step, the score in the current cell is determine by the scores in the neighboring cells
  - The maximal score and the direction that gave that score is stored (we will use these later to determine the optimal alignment)



				j		
		-	D	P	L	E
	-	0	-2	-4	-6	-8
i	D	-2	1	-1		
	P	-4				
	M	-6				
	E	-8				

Scores: match = +1, mismatch = -1, gap = -2

↘ ① (-2)+(-1) = -3 <= (D-P) mismatch!

↓ ② (-4)+(-2) = -6

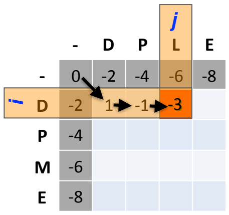
→ ③ (1)+(-2) = -1

Alignment

```
D-
DP
```

Scoring the alignment matrix

- We will continue to store the alignment score ($S_{i,j}$) for all possible alignments in the alignment matrix.



					j	
		-	D	P	L	E
	-	0	-2	-4	-6	-8
i	D	-2	1	-1	-3	
	P	-4				
	M	-6				
	E	-8				

Scores: match = +1, mismatch = -1, gap = -2

↘ ① (-4)+(-1) = -5 <= (D-L) mismatch

↓ ② (-6)+(-2) = -8

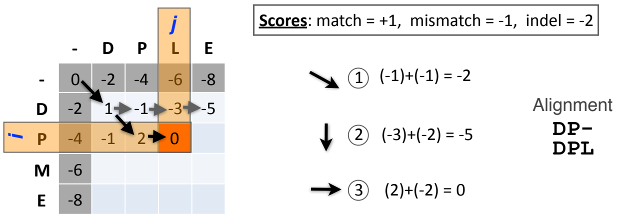
→ ③ (-1)+(-2) = -3

Alignment

```
D--
DPL
```

Scoring the alignment matrix

- For the highlighted cell, the corresponding score ($S_{i,j}$) refers to the score of the optimal alignment of the first *i* characters from sequence1, and the first *j* characters from sequence2.



	-	D	P	L	E
-	0	-2	-4	-6	-8
D	-2	1	-1	-3	-5
P	-4	-1	2	0	
M	-6				
E	-8				

Scores: match = +1, mismatch = -1, indel = -2

↘ ① (-1)+(-1) = -2

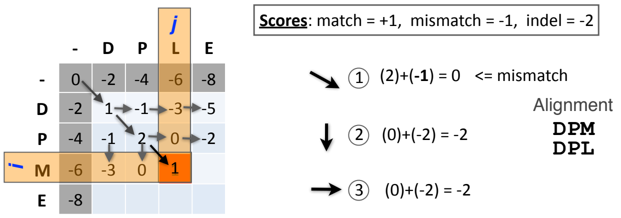
↓ ② (-3)+(-2) = -5

→ ③ (2)+(-2) = 0

Alignment
```
DP-
DPL
```

Scoring the alignment matrix

- At each step, the score in the current cell is determine by the scores in the neighboring cells
  - The maximal score and the direction that gave that score is stored



	-	D	P	L	E
-	0	-2	-4	-6	-8
D	-2	1	-1	-3	-5
P	-4	-1	2	0	-2
M	-6	-3	0	1	
E	-8				

Scores: match = +1, mismatch = -1, indel = -2

↘ ① (2)+(**-1**) = 0 <= mismatch

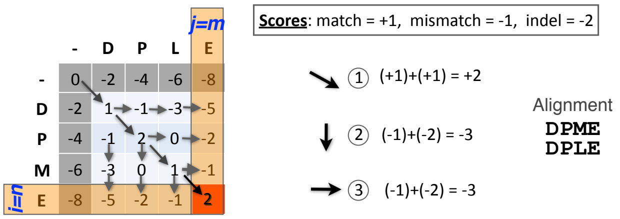
↓ ② (0)+(-2) = -2

→ ③ (0)+(-2) = -2

Alignment
```
DPM
DPL
```

Scoring the alignment matrix

- The score of the best alignment of the entire sequences corresponds to $S_{n,m}$
  - (where *n* and *m* are the length of the sequences)



	-	D	P	L	E
-	0	-2	-4	-6	-8
D	-2	1	-1	-3	-5
P	-4	-1	2	0	-2
M	-6	-3	0	1	-1
E	-8	-5	-2	-1	2

Scores: match = +1, mismatch = -1, indel = -2

↘ ① (+1)+(+1) = +2

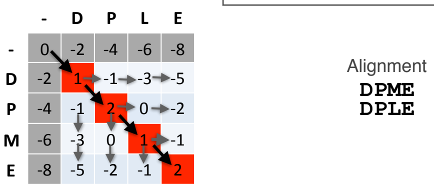
↓ ② (-1)+(-2) = -3

→ ③ (-1)+(-2) = -3

Alignment
```
DPME
DPLE
```

Scoring the alignment matrix

- To find the best alignment, we retrace the arrows starting from the bottom right cell
  - N.B. The optimal alignment score and alignment are dependent on the chosen scoring system



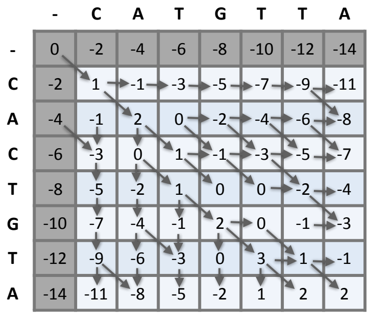
	-	D	P	L	E
-	0	-2	-4	-6	-8
D	-2	1	-1	-3	-5
P	-4	-1	2	0	-2
M	-6	-3	0	1	-1
E	-8	-5	-2	-1	2

Scores: match = +1, mismatch = -1, indel = -2

Alignment
```
DPME
DPLE
```

Questions:

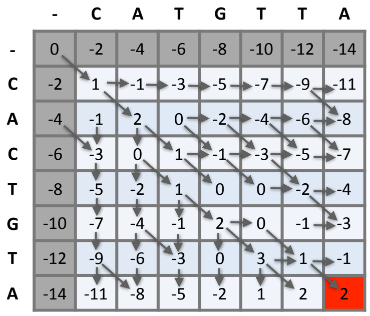
- What is the optimal score for the alignment of these sequences and how do we find the optimal alignment?



	-	C	A	T	G	T	T	A
-	0	-2	-4	-6	-8	-10	-12	-14
C	-2	1	-1	-3	-5	-7	-9	-11
A	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	-1	-3	-5	-7
T	-8	-5	-2	1	0	0	-2	-4
G	-10	-7	-4	-1	2	0	-1	-3
T	-12	-9	-6	-3	0	3	1	-1
A	-14	-11	-8	-5	-2	1	2	2

Questions:

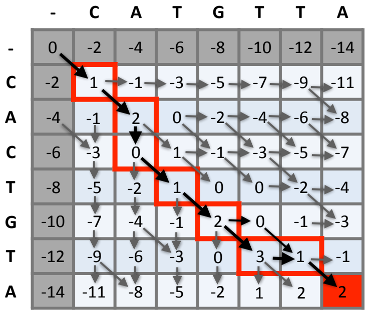
- What is the optimal score for the alignment of these sequences and how do we find the optimal alignment?



	-	C	A	T	G	T	T	A
-	0	-2	-4	-6	-8	-10	-12	-14
C	-2	1	-1	-3	-5	-7	-9	-11
A	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	-1	-3	-5	-7
T	-8	-5	-2	1	0	0	-2	-4
G	-10	-7	-4	-1	2	0	-1	-3
T	-12	-9	-6	-3	0	3	1	-1
A	-14	-11	-8	-5	-2	1	2	2

Questions:

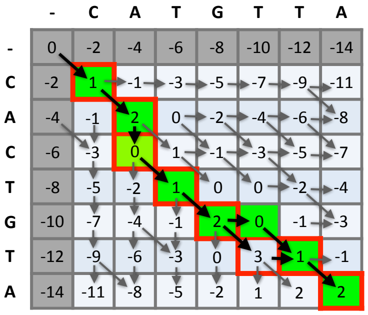
- To find the best alignment we retrace the arrows starting from the bottom right cell



	-	C	A	T	G	T	T	A
-	0	-2	-4	-6	-8	-10	-12	-14
C	-2	1	-1	-3	-5	-7	-9	-11
A	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	-1	-3	-5	-7
T	-8	-5	-2	1	0	0	-2	-4
G	-10	-7	-4	-1	2	0	-1	-3
T	-12	-9	-6	-3	0	3	1	-1
A	-14	-11	-8	-5	-2	1	2	2

More than one alignment possible

- Sometimes more than one alignment can result in the same optimal score



	-	C	A	T	G	T	T	A
-	0	-2	-4	-6	-8	-10	-12	-14
C	-2	1	-1	-3	-5	-7	-9	-11
A	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	-1	-3	-5	-7
T	-8	-5	-2	1	0	0	-2	-4
G	-10	-7	-4	-1	2	0	-1	-3
T	-12	-9	-6	-3	0	3	1	-1
A	-14	-11	-8	-5	-2	1	2	2

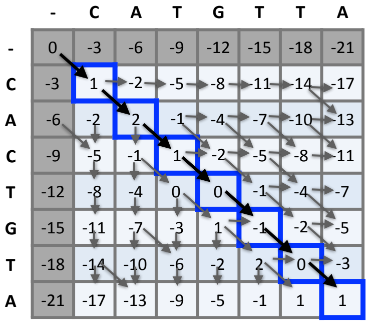
Alignment

```
CACTGT-A
CA-TGTTA

CACTG-TA
CA-TGTTA
```

The alignment and score are dependent on the scoring system

- Here we increase the gap penalty from -2 to -3



	-	C	A	T	G	T	T	A
-	0	-3	-6	-9	-12	-15	-18	-21
C	-3	1	-2	-5	-8	-11	-14	-17
A	-6	-2	2	-1	-4	-7	-10	-13
C	-9	-5	-1	1	-2	-5	-8	-11
T	-12	-8	-4	0	0	-1	-4	-7
G	-15	-11	-7	-3	1	-1	-2	-5
T	-18	-14	-10	-6	-2	2	0	-3
A	-21	-17	-13	-9	-5	-1	1	1

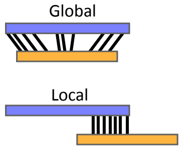
Alignment

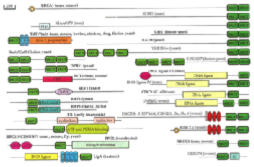
```
CACTGT-A
CA-TGTTA

CACTG-TA
CA-TGTTA
- - - - - - - - - - - -
CACTGTA
CATGTTA
```

Global *vs* local alignments

- Needleman-Wunsch is a **global alignment** algorithm
  - Resulting alignment spans the complete sequences end to end
  - This is appropriate for closely related sequences that are similar in length
- For many practical applications we require **local alignments**
  - Local alignments highlight sub-regions (*e.g.* protein domains) in the two sequences that align well





Local alignment: Definition

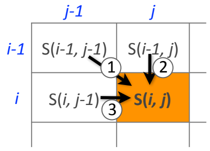
- Smith & Waterman proposed simply that a local alignment of two sequences allow arbitrary-length segments of each sequence to be aligned, with no penalty for the unaligned portions of the sequences. Otherwise, the score for a local alignment is calculated the same way as that for a global alignment

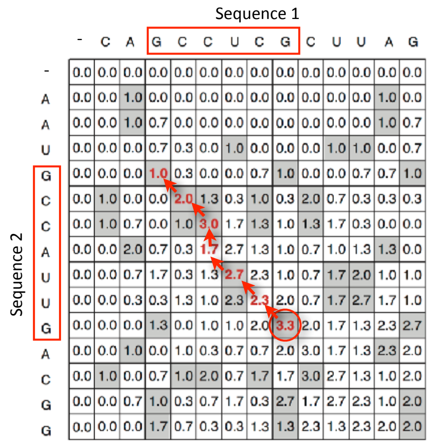
Smith, T.F. & Waterman, M.S. (1981) "Identification of common molecular subsequences." J. Mol. Biol. 147:195-197.

The Smith-Waterman algorithm

- Three main modifications to Needleman-Wunsch:
  - Allow a node to start at 0
  - The score for a particular cell cannot be negative
    - if all other score options produce a negative value, then a zero must be inserted in the cell
  - Record the highest- scoring node, and trace back from there

$$S(i, j) = \text{Max} \begin{cases} S(i\text{-}1, j\text{-}1) + \text{(mis)match} & \searrow \; ① \\ S(i\text{-}1, j) \text{ - gap penalty} & \downarrow \; ② \\ S(i, j\text{-}1) \text{ - gap penalty} & \rightarrow \; ③ \\ \mathbf{0} & ④ \end{cases}$$





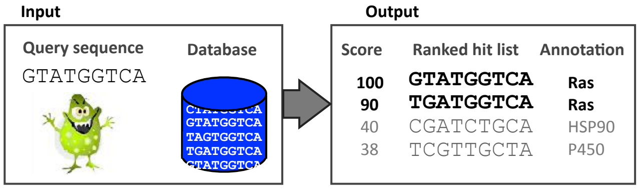
	-	C	A	G	C	C	U	C	G	C	U	U	A	G
-	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
A	0.0	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.7
U	0.0	0.0	0.0	0.7	0.3	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.7
G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	0.7	0.7	1.0
C	0.0	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3	0.3
C	0.0	1.0	0.7	0.0	1.0	3.0	1.7	1.3	1.0	1.3	1.7	0.3	0.0	0.0
A	0.0	0.0	2.0	0.7	0.3	1.7	2.7	1.3	1.0	0.7	1.0	1.3	1.3	0.0
U	0.0	0.0	0.7	1.7	0.3	1.3	2.7	2.3	1.0	0.7	1.7	2.0	1.0	1.0
U	0.0	0.0	0.3	0.3	1.3	1.0	2.3	2.3	2.0	0.7	1.7	2.7	1.7	1.0
G	0.0	0.0	0.0	1.3	0.0	1.0	1.0	2.0	3.3	2.0	1.7	1.3	2.3	2.7
A	0.0	0.0	1.0	0.0	1.0	0.3	0.7	0.7	2.0	3.0	1.7	1.3	2.3	2.0
C	0.0	1.0	0.0	0.7	1.0	2.0	0.7	1.7	1.7	3.0	2.7	1.3	1.0	2.0
G	0.0	0.0	0.7	1.0	0.3	0.7	1.7	0.3	2.7	1.7	2.7	2.3	1.0	2.0
G	0.0	0.0	0.0	1.7	0.7	0.3	0.3	1.3	1.3	2.3	1.3	2.3	2.0	2.0

Local alignment

```
GCC-AUG
GCCUCGC
```

Local alignments can be used for database searching

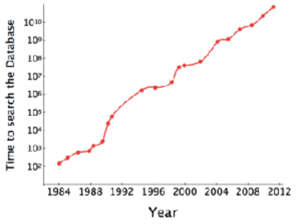
- **Goal**: Given a query sequence (Q) and a sequence database (D), find a list of sequences from D that are most similar to Q
  - **Input**: Q, D and scoring scheme
  - **Output**: Ranked list of hits



Score	Ranked hit list	Annotation
100	GTATGGTCA	Ras
90	TGATGGTCA	Ras
40	CGATCTGCA	HSP90
38	TCGTTGCTA	P450

The database search problem

- Due to the rapid growth of sequence databases, search algorithms have to be both efficient and sensitive
  - Time to search with SW is proportional to *m* x *n* (*m* is length of query, n is length of database), **too slow for large databases**!

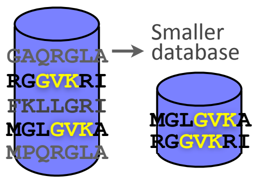


To reduce search time **heuristic algorithms**, such as BLAST, first remove database sequences without a strong local similarity to the query sequence in a quick initial scan.

The database search problem

- Due to the rapid growth of sequence databases, search algorithms have to be both efficient and sensitive
  - Time to search with SW is proportional to *m* x *n* (*m* is length of query, n is length of database), **too slow for large databases**!

Query **RGGVKRIKLMR**



To reduce search time **heuristic algorithms**, such as BLAST, first remove database sequences without a strong local similarity to the query sequence in a quick initial scan.

Outline for today

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing
    (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - PSI-BLAST and HMM approaches

Rapid, heuristic versions of Smith–Waterman: **BLAST**

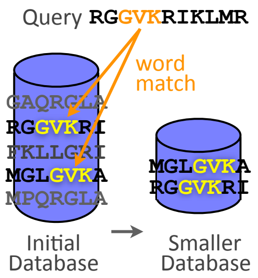
- BLAST (Basic Local Alignment Search Tool) is a simplified form of Smith-Waterman (SW) alignment that is popular because it is **fast** and **easily accessible**
  - BLAST is a heuristic approximation to SW - It examines only part of the search space
  - BLAST saves time by restricting the search by scanning database sequences for likely matches before performing more rigorous alignments
  - Sacrifices some sensitivity in exchange for speed
  - In contrast to SW, BLAST is not guaranteed to find optimal alignments

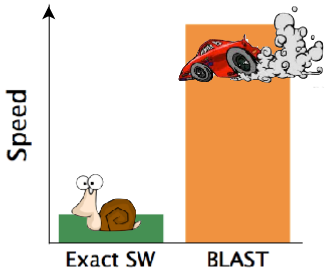
Rapid, heuristic versions of Smith–Waterman: **BLAST**

- BLAST (Basic Local Alignment Search Tool) i… …ied form of Smith-Waterman (SW) alig… …ar because it is **fast** and **easily** …
  - BLAST finds region… …gical sequences
  - BLA… …e search by scanning … …kely matches before performing … …gnments
  - … some sensitivity in exchange for speed
  - …n contrast to SW, BLAST is not guaranteed to find optimal alignments

“The central idea of the BLAST algorithm is to confine attention to sequence pairs that contain an initial **word pair** match”
Altschul et al. (1990)

- BLAST uses this pre-screening heuristic approximation resulting in an an approach that is about 50 times faster than the Smith-Waterman algorithm





How BLAST works

- Four basic phases
  - **Phase 1**: compile a list of query word pairs (w=3)

```
                   RGGVKRI   Query sequence
                   RGG
                    GGV
generate list of     GVK
w=3 words for         VKR
query                  KRI
```

Blast

- **Phase 2**: expand word pairs to include those similar to query (defined as those above a similarity threshold to original word, i.e. match scores in substitution matrix)

```
                   RGGVKRI   Query sequence
                   RGG RAG RIG RLG  . . .
                    GGV GAV GTV GCV  . . .
extend list of       GVK GAK GIK GGK  . . .
words similar         VKR VRR VHR VER  . . .
to query               KRI KKI KHI KDI  . . .
```

Blast

- **Phase 3**: a database is scanned to find sequence entries that match the compiled word list

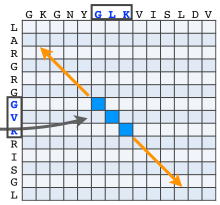
```
GNYGLKVISLDVE   Database sequence
. . . . . . . . . . . . . . . . . . . . . . . .
                     RGGVKRI   Query sequence
                     RGG RAG RIG RLG  . . .
search for            GGV GAV GTV GCV  . . .
perfect matches        GVK GLK GIK GGK  . . .
in the database         VKR VRR VHR VER  . . .
sequence                 KRI KKI KHI KDI  . . .
```

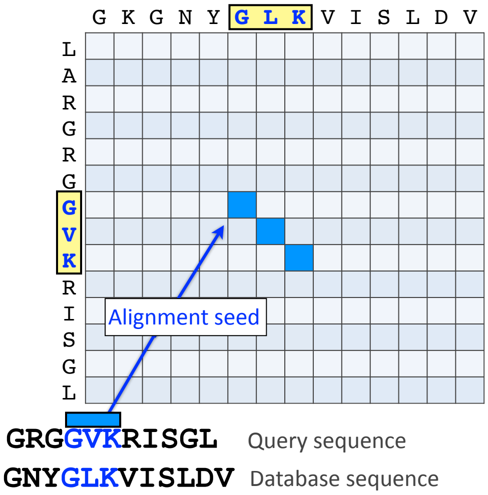
Blast

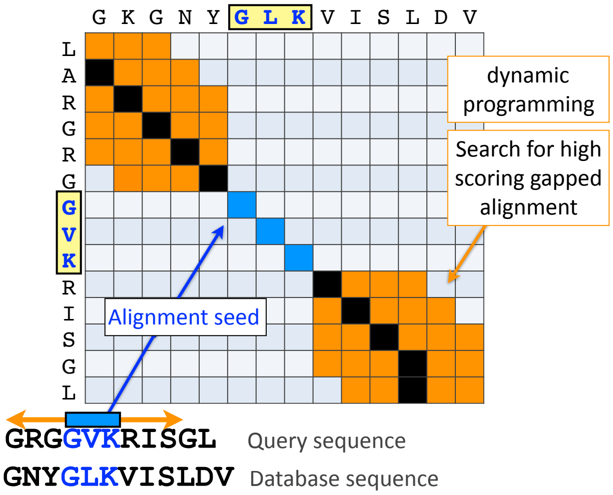
- **Phase 4**: the initial database hits are extended in both directions using dynamic programing

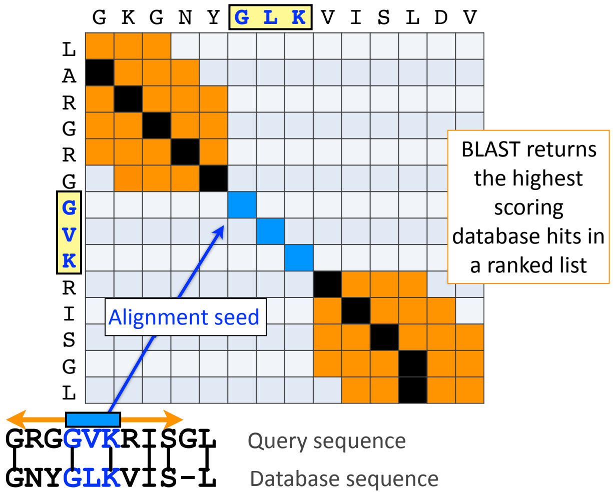
```
GNYGLKVISLDVE   Database sequence
. . . . . . . . . . . . . . . . . . . . . . . .
  RGGVKRI              Query sequence
```

matched word is used as a local **alignment seed**






G K G N Y G L K V I S L D V
L A R G R G G V K R I S G L
dynamic programming
Search for high scoring gapped alignment
Alignment seed
GRGGVKRISGL
Query sequence
GNYGLKVISLDV
Database sequence


G K G N Y G L K V I S L D V
L A R G R G G V K R I S G L
BLAST returns the highest scoring database hits in a ranked list
Alignment seed
GRGGVKRISGL
Query sequence
GNYGLKVIS-L
Database sequence

BLAST output

- BLAST returns the highest scoring database hits in a ranked list along with details about the target sequence and alignment statistics

Description	Max score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo sapiens]	677	100%	0	100%	NP_004512.1
Kif5b protein [Mus musculus]	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	48.2	40%	0.03	32%	ELK35081.1
mKIAA4102 protein [Mus musculus]	42.7	38%	3.02	24%	EHH28205.1

Statistical significance of results

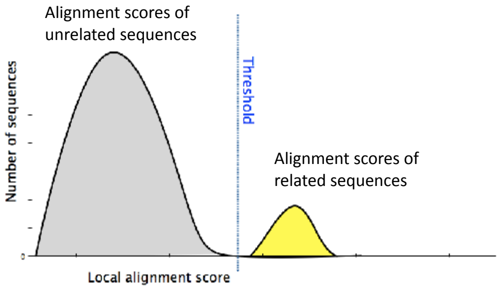
- An important feature of BLAST is the computation of statistical significance for each hit. This is described by the **E value** (expect value)

Description	Max score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo sapiens]	677	100%	0	100%	NP_004512.1
Kif5b protein [Mus musculus]	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	48.2	40%	0.03	32%	ELK35081.1
mKIAA4102 protein [Mus musculus]	42.7	38%	3.02	24%	EHH28205.1

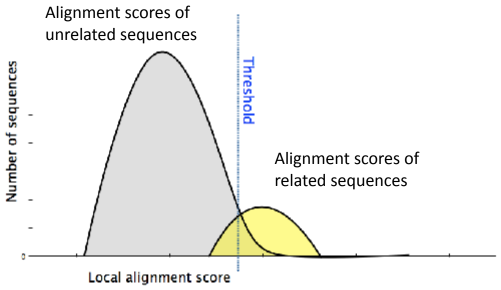
BLAST scores and E-values

- The **E value** is the **expected** number of hits that are as good or better than the observed local alignment score (with this score or better) if the query and database are **random** with respect to each other
  - *i.e.* the number of alignments expected to occur by chance with equivalent or better scores
- Typically, only hits with E value **below** a significance threshold are reported
  - This is equivalent to selecting alignments with score above a certain score threshold

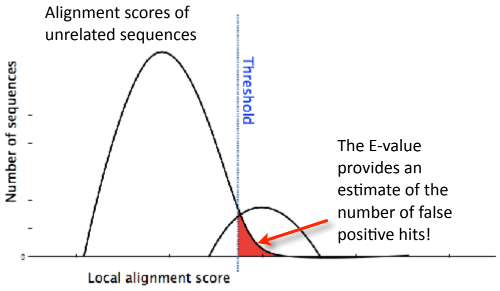
- Ideally, a threshold separates all query related sequences (yellow) from all unrelated sequences (gray)



- Unfortunately, often both score distributions overlap
  - The E value describes the expected number of hits with a score above the threshold if the query and database are unrelated

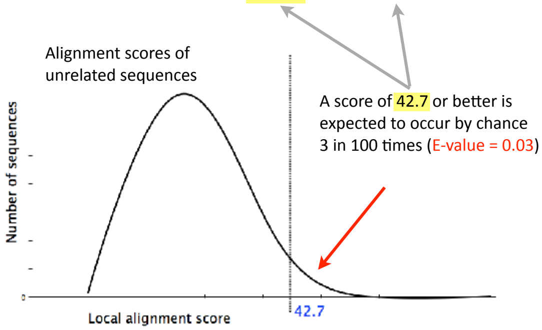


- Unfortunately, often both score distributions overlap
  - The E value describes the expected number of hits with a score above the threshold if the query and database are unrelated



The E-value provides an estimate of the number of false positive hits!

Description	Max score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo sapiens]	677	100%	0	100%	NP_004512.1
Kif5b protein [Mus musculus]	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	42.7	40%	**0.03**	32%	ELK35081.1



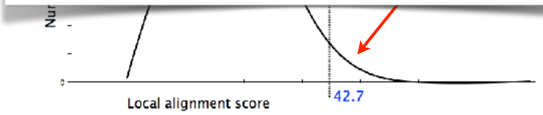
A score of 42.7 or better is expected to occur by chance 3 in 100 times (E-value = 0.03)

Description	Max score	Total score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo	677	677	100%	0	100%	NP_004512.1

In general *E* values < 0.005 are usually significant.

To find out more about *E* values see: “*The Statistics of Sequence Similarity Scores*” available in the help section of the NCBI BLAST site:

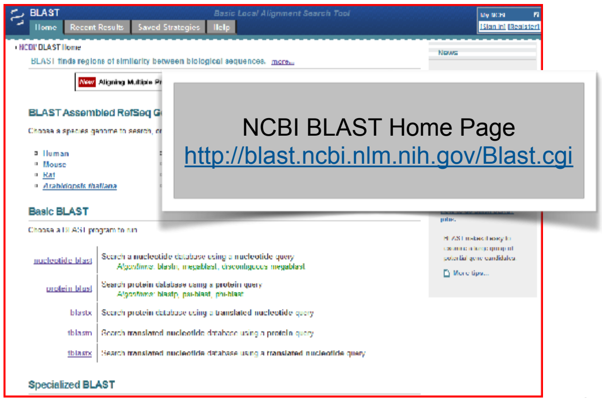
http://www.ncbi.nlm.nih.gov/blast/tutorial/Altschul-1.html



Outline for today

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing
  (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - BLAST, PSI-BLAST and HMM approaches

Practical database searching with BLAST



BLAST Basic Local Alignment Search Tool

NCBI/ BLAST Home

BLAST finds regions of similarity between biological sequences. more...

New Aligning Multiple Pr

BLAST Assembled RefSeq G

Choose a species genome to search, or

- Human
- Mouse
- Rat
- *Arabidopsis thaliana*

Basic BLAST

Choose a BLAST program to run

nucleotide blast	Search a nucleotide database using a nucleotide query Algorithms: blastn, megablast, discontiguous megablast
protein blast	Search protein database using a protein query Algorithms: blastp, psi-blast, phi-blast
blastx	Search protein database using a translated nucleotide query
tblastn	Search translated nucleotide database using a protein query
tblastx	Search translated nucleotide database using a translated nucleotide query

BLAST makes it easy to examine a large group of potential gene candidates

More tips...

Specialized BLAST

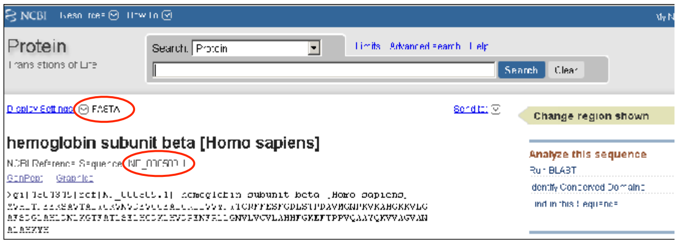
NCBI BLAST Home Page
http://blast.ncbi.nlm.nih.gov/Blast.cgi

Practical database searching with BLAST

- There are four basic components to a traditional BLAST search
  - (1) Choose the sequence (query)
  - (2) Select the BLAST program
  - (3) Choose the database to search
  - (4) Choose optional parameters
- Then click "BLAST"

Step 1: Choose your sequence

- Sequence can be input in FASTA format or as accession number



hemoglobin subunit beta [Homo sapiens]

NCBI Reference Sequence: NP_000509.1

GenPept Graphics

```
>gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLG
AFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVAN
ALAHKYH
```

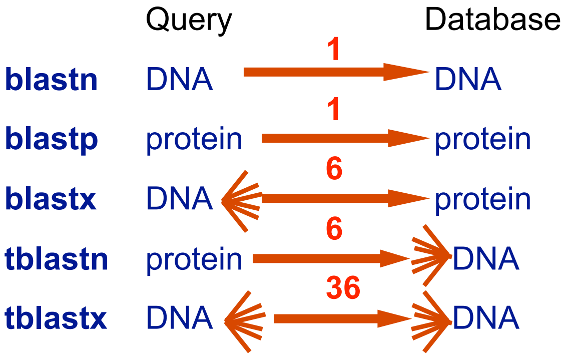
Analyze this sequence

Run BLAST

Identify Conserved Domains

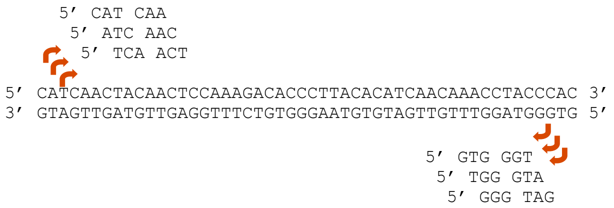
Find in this Sequence

Step 2: Choose the BLAST program



	Query		Database
blastn	DNA	1	DNA
blastp	protein	1	protein
blastx	DNA	6	protein
tblastn	protein	6	DNA
tblastx	DNA	36	DNA

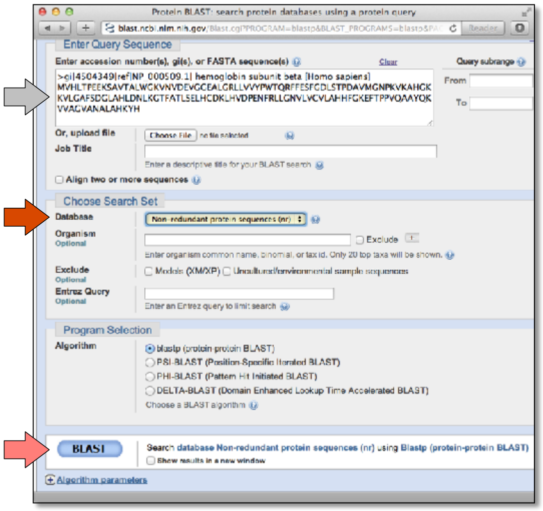
DNA potentially encodes six proteins



```
      5’ CAT CAA
        5’ ATC AAC
          5’ TCA ACT

5’ CATCAACTACAACTCCAAAGACACCCTTACACATCAACAAACCTACCCAC 3’
3’ GTAGTTGATGTTGAGGTTTCTGTGGGAATGTGTAGTTGTTTGGATGGGTG 5’

                                          5’ GTG GGT
                                            5’ TGG GTA
                                              5’ GGG TAG
```



Protein BLAST: search protein databases using a protein query

blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&BLAST_PROGRAMS=blastp&PA

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

```
>gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGK
KVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQK
VVAGVANALAHKYH
```

Or, upload file: Choose File no file selected

Job Title: Enter a descriptive title for your BLAST search

□ Align two or more sequences

Choose Search Set

Database: Non-redundant protein sequences (nr)

Organism (Optional): □ Exclude — Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.

Exclude (Optional): □ Models (XM/XP) □ Uncultured/environmental sample sequences

Entrez Query (Optional): Enter an Entrez query to limit search

Program Selection

Algorithm:

- ● blastp (protein-protein BLAST)
- ○ PSI-BLAST (Position-Specific Iterated BLAST)
- ○ PHI-BLAST (Pattern Hit Initiated BLAST)
- ○ DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)

Choose a BLAST algorithm

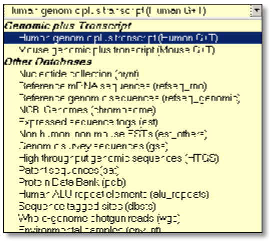
BLAST — Search database Non-redundant protein sequences (nr) using Blastp (protein-protein BLAST)

□ Show results in a new window

⊕ Algorithm parameters

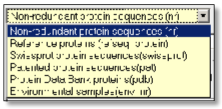
Step 3: Choose the database

nr = non-redundant (most general database)
dbest = database of expressed sequence tags
dbsts = database of sequence tag sites
gss = genomic survey sequences



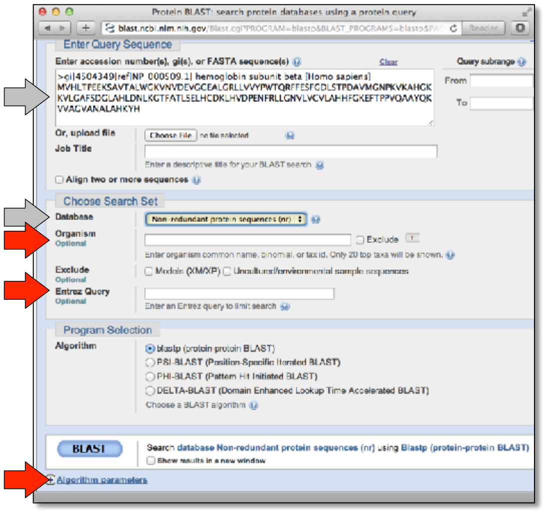
Human genomic plus transcript (Human G+T)
Genomic plus Transcript
Human genomic plus transcript (Human G+T)
Mouse genomic plus transcript (Mouse G+T)
Other Databases
Nucleotide collection (nr/nt)
Reference mRNA sequences (refseq_rna)
Reference genomic sequences (refseq_genomic)
NCBI Genomes (chromosome)
Expressed sequence tags (est)
Non-human, non-mouse ESTs (est_others)
Genomic survey sequences (gss)
High throughput genomic sequences (HTGS)
Patent sequences (pat)
Protein Data Bank (pdb)
Human ALU repeat elements (alu_repeats)
Sequence tagged sites (dbsts)
Whole-genome shotgun reads (wgs)
Environmental samples (env_nt)

nucleotide databases



Non-redundant protein sequences (nr)
Non-redundant protein sequences (nr)
Reference proteins (refseq_protein)
Swissprot protein sequences(swissprot)
Patented protein sequences(pat)
Protein Data Bank proteins(pdb)
Environmental samples(env_nr)

protein databases



Protein BLAST: search protein databases using a protein query
blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&BLAST_PROGRAMS=blastp&PA

Enter Query Sequence
Enter accession number(s), gi(s), or FASTA sequence(s)   Clear   Query subrange
>gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGK
KVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQK
VVAGVANALAHKYH
From
To
Or, upload file   Choose File   no file selected
Job Title
Enter a descriptive title for your BLAST search
☐ Align two or more sequences

Choose Search Set
Database   Non-redundant protein sequences (nr)
Organism Optional   ☐ Exclude
Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.
Exclude Optional   ☐ Models (XM/XP) ☐ Uncultured/environmental sample sequences
Entrez Query Optional
Enter an Entrez query to limit search

Program Selection
Algorithm
● blastp (protein-protein BLAST)
○ PSI-BLAST (Position-Specific Iterated BLAST)
○ PHI-BLAST (Pattern Hit Initiated BLAST)
○ DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)
Choose a BLAST algorithm

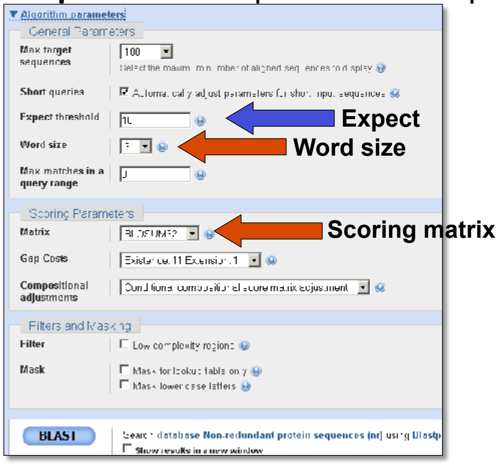
BLAST   Search database Non-redundant protein sequences (nr) using Blastp (protein-protein BLAST)
☐ Show results in a new window
⊕ Algorithm parameters

Organism

Entrez

Settings!

Step 4a: Select optional search parameters

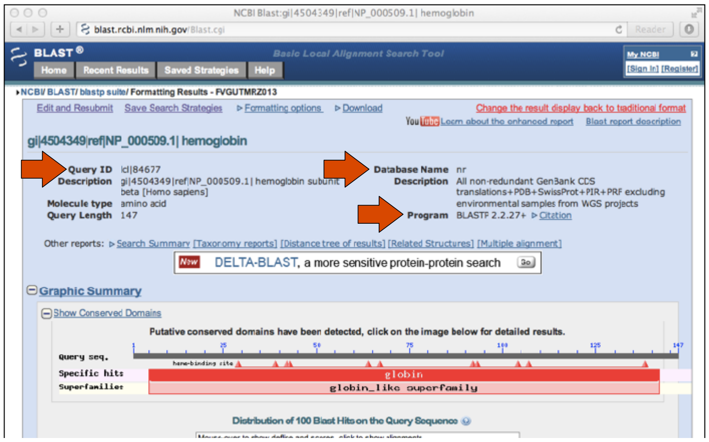


▼ Algorithm parameters
General Parameters
Max target sequences   100
Select the maximum number of aligned sequences to display
Short queries   ☑ Automatically adjust parameters for short input sequences
Expect threshold   10
Word size   3
Max matches in a query range   0
Scoring Parameters
Matrix   BLOSUM62
Gap Costs   Existence: 11 Extension: 1
Compositional adjustments   Conditional compositional score matrix adjustment
Filters and Masking
Filter   ☐ Low complexity regions
Mask   ☐ Mask for lookup table only
☐ Mask lower case letters
BLAST   Search database Non-redundant protein sequences (nr) using Blastp
☐ Show results in a new window

Step 4: Optional parameters

- You can…
  - – choose the organism to search
  - – change the substitution matrix
  - – change the expect (E) value
  - – change the word size
  - – change the output format

Results page



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin
blast.ncbi.nlm.nih.gov/Blast.cgi

▸NCBI/ BLAST/ blastp suite/ Formatting Results - FVGUTMRZ013
Edit and Resubmit   Save Search Strategies   ▹Formatting options   ▹Download   Change the result display back to traditional format
YouTube Learn about the enhanced report   Blast report description

gi|4504349|ref|NP_000509.1| hemoglobin

Query ID   lcl|84677
Description   gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo sapiens]
Molecule type   amino acid
Query Length   147

Database Name   nr
Description   All non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF excluding environmental samples from WGS projects
Program   BLASTP 2.2.27+ ▹Citation

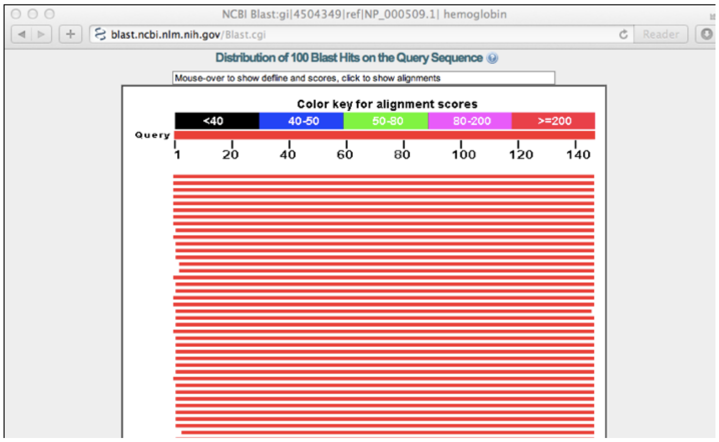
Other reports: ▹Search Summary [Taxonomy reports] [Distance tree of results] [Related Structures] [Multiple alignment]

New DELTA-BLAST, a more sensitive protein-protein search   Go

⊖ Graphic Summary
⊖ Show Conserved Domains
Putative conserved domains have been detected, click on the image below for detailed results.

Distribution of 100 Blast Hits on the Query Sequence
Mouse-over to show define and scores, click to show alignments

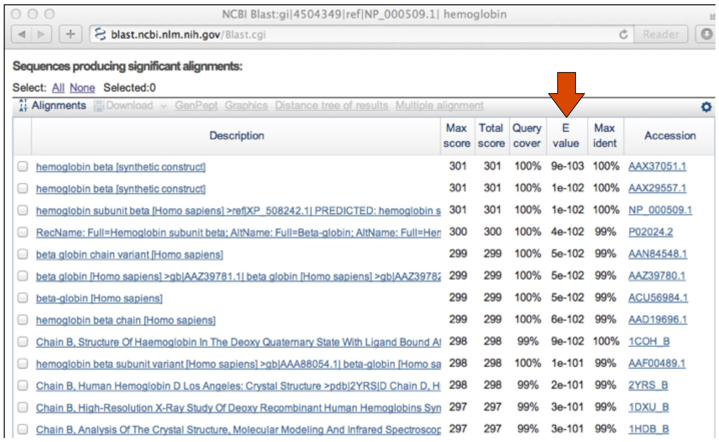
Further down the results page…



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin
blast.ncbi.nlm.nih.gov/Blast.cgi

Distribution of 100 Blast Hits on the Query Sequence
Mouse-over to show define and scores, click to show alignments

Further down the results page...



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin

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

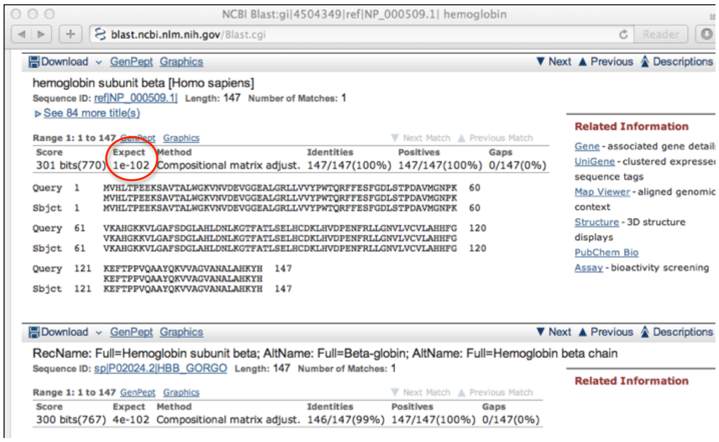
Sequences producing significant alignments:

Select: All None Selected:0

Alignments Download GenPept Graphics Distance tree of results Multiple alignment

	Description	Max score	Total score	Query cover	E value	Max ident	Accession
□	hemoglobin beta [synthetic construct]	301	301	100%	9e-103	100%	AAX37051.1
□	hemoglobin beta [synthetic construct]	301	301	100%	1e-102	100%	AAX29557.1
□	hemoglobin subunit beta [Homo sapiens] >ref|XP_508242.1| PREDICTED: hemoglobin s	301	301	100%	1e-102	100%	NP_000509.1
□	RecName: Full=Hemoglobin subunit beta; AltName: Full=Beta-globin; AltName: Full=Hem	300	300	100%	4e-102	99%	P02024.2
□	beta globin chain variant [Homo sapiens]	299	299	100%	5e-102	99%	AAN84548.1
□	beta globin [Homo sapiens] >gb|AAZ39781.1| beta globin [Homo sapiens] >gb|AAZ3978	299	299	100%	5e-102	99%	AAZ39780.1
□	beta-globin [Homo sapiens]	299	299	100%	5e-102	99%	ACU56984.1
□	hemoglobin beta chain [Homo sapiens]	299	299	100%	6e-102	99%	AAD19696.1
□	Chain B, Structure Of Haemoglobin In The Deoxy Quaternary State With Ligand Bound At	298	298	99%	9e-102	100%	1COH_B
□	hemoglobin beta subunit variant [Homo sapiens] >gb|AAA88054.1| beta-globin [Homo sa	298	298	100%	1e-101	99%	AAF00489.1
□	Chain B, Human Hemoglobin D Los Angeles: Crystal Structure >pdb|2YRS|D Chain D, H	298	298	99%	2e-101	99%	2YRS_B
□	Chain B, High-Resolution X-Ray Study Of Deoxy Recombinant Human Hemoglobins Syn	297	297	99%	3e-101	99%	1DXU_B
□	Chain B, Analysis Of The Crystal Structure, Molecular Modeling And Infrared Spectroscop	297	297	99%	3e-101	99%	1HDB_B

Further down the results page...



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin

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

Download GenPept Graphics ▼ Next ▲ Previous Descriptions

hemoglobin subunit beta [Homo sapiens]

Sequence ID: ref|NP_000509.1| Length: 147 Number of Matches: 1

▷ See 84 more title(s)

Range 1: 1 to 147 GenPept Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps
301 bits(770)	1e-102	Compositional matrix adjust.	147/147(100%)	147/147(100%)	0/147(0%)

```
Query  1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
            MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK
Sbjct  1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60

Query  61   VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG  120
            VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG
Sbjct  61   VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG  120

Query  121  KEFTPPVQAAYQKVVAGVANALAHKYH  147
            KEFTPPVQAAYQKVVAGVANALAHKYH
Sbjct  121  KEFTPPVQAAYQKVVAGVANALAHKYH  147
```

Related Information

Gene - associated gene detail

UniGene - clustered expressed sequence tags

Map Viewer - aligned genomic context

Structure - 3D structure displays

PubChem Bio

Assay - bioactivity screening

Download GenPept Graphics ▼ Next ▲ Previous Descriptions

RecName: Full=Hemoglobin subunit beta; AltName: Full=Beta-globin; AltName: Full=Hemoglobin beta chain

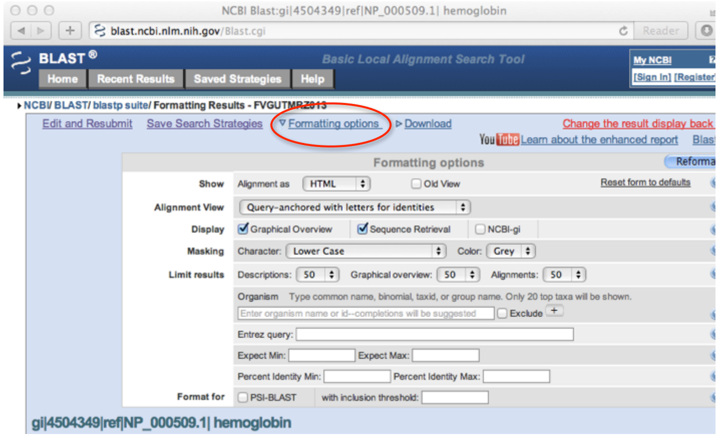
Sequence ID: sp|P02024.2|HBB_GORGO Length: 147 Number of Matches: 1

Related Information

Range 1: 1 to 147 GenPept Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps
300 bits(767)	4e-102	Compositional matrix adjust.	146/147(99%)	147/147(100%)	0/147(0%)

Different output formats are available



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin

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

▸ NCBI/ BLAST/ blastp suite/ Formatting Results - FVGUTMRZ013

Edit and Resubmit   Save Search Strategies   ▽ Formatting options   ▷ Download   Change the result display back

YouTube Learn about the enhanced report   Blas

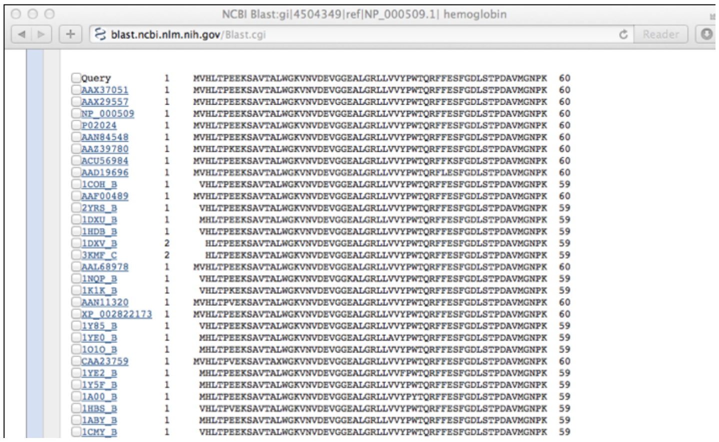
Formatting options   Reforma

Reset form to defaults

- Show: Alignment as HTML; □ Old View
- Alignment View: Query-anchored with letters for identities
- Display: ☑ Graphical Overview ☑ Sequence Retrieval □ NCBI-gi
- Masking: Character: Lower Case; Color: Grey
- Limit results: Descriptions: 50; Graphical overview: 50; Alignments: 50
- Organism: Type common name, binomial, taxid, or group name. Only 20 top taxa will be shown. Enter organism name or id--completions will be suggested □ Exclude +
- Entrez query:
- Expect Min: Expect Max:
- Percent Identity Min: Percent Identity Max:
- Format for: □ PSI-BLAST with inclusion threshold:

gi|4504349|ref|NP_000509.1| hemoglobin

E.g. Query anchored alignments

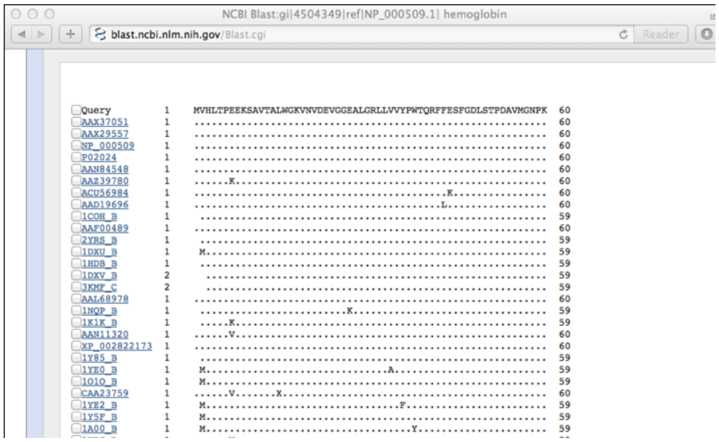


NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin

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

```
□Query          1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□AAX37051       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□AAX29557       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□NP_000509      1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□P02024         1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□AAN84548       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□AAZ39780       1    MVHLTPKEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□ACU56984       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFKSFGDLSTPDAVMGNPK  60
□AAD19696       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFLESFGDLSTPDAVMGNPK  60
□1COH_B         1     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□AAF00489       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□2YRS_B         1     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1DXU_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1HDB_B         1     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1DXV_B         2      HLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□3KMF_C         2      HLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□AAL68978       1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□1NQP_B         1     VHLTPEEKSAVTALWGKVNVDEVGGKALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1K1K_B         1     VHLTPKEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□AAN11320       1    MVHLTPVEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□XP_002822173   1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□1Y85_B         1     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1YE0_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLAVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1O1O_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□CAA23759       1    MVHLTPVEKSAVTAXWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□1YE2_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVFPWTQRFFESFGDLSTPDAVMGNPK  59
□1Y5F_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1A00_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPYTQRFFESFGDLSTPDAVMGNPK  59
□1HBS_B         1     VHLTPVEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1ABY_B         1     MHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
□1CMY_B         1     VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  59
```

... and alignments with dots for identities



NCBI Blast:gi|4504349|ref|NP_000509.1| hemoglobin

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

```
□Query          1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK  60
□AAX37051       1    ............................................................  60
□AAX29557       1    ............................................................  60
□NP_000509      1    ............................................................  60
□P02024         1    ............................................................  60
□AAN84548       1    ............................................................  60
□AAZ39780       1    ......K.....................................................  60
□ACU56984       1    ...........................................K................  60
□AAD19696       1    ..........................................L.................  60
□1COH_B         1     ...........................................................  59
□AAF00489       1    ............................................................  60
□2YRS_B         1     ...........................................................  59
□1DXU_B         1     M..........................................................  59
□1HDB_B         1     ...........................................................  59
□1DXV_B         2      ..........................................................  59
□3KMF_C         2      ..........................................................  59
□AAL68978       1    ............................................................  60
□1NQP_B         1     .........................K.................................  59
□1K1K_B         1     .....K.....................................................  59
□AAN11320       1    ......V.....................................................  60
□XP_002822173   1    ............................................................  60
□1Y85_B         1     ...........................................................  59
□1YE0_B         1     M...............................A..........................  59
□1O1O_B         1     M..........................................................  59
□CAA23759       1    ......V.......X.............................................  60
□1YE2_B         1     M.................................F........................  59
□1Y5F_B         1     M..........................................................  59
□1A00_B         1     M...................................Y......................  59
```

Common problems

- Selecting the wrong version of BLAST
- Selecting the wrong database
- Too many hits returned
- Too few hits returned
- Unclear about the significance of a particular result - are these sequences homologous?

How to handle too many results

- Focus on the question you are trying to answer
  - – select “refseq” database to eliminate redundant matches from “nr”
  - – Limit hits by organism
  - – Use just a portion of the query sequence, when appropriate
  - – Adjust the expect value; lowering *E* will reduce the number of matches returned

How to handle too few results

- Many genes and proteins have no significant database matches
  - – remove Entrez limits
  - – raise E-value threshold
  - – search different databases
  - – try scoring matrices with lower BLOSUM values (or higher PAM values)
  - – use a search algorithm that is more sensitive than BLAST (*e.g.* PSI-BLAST or HMMer)

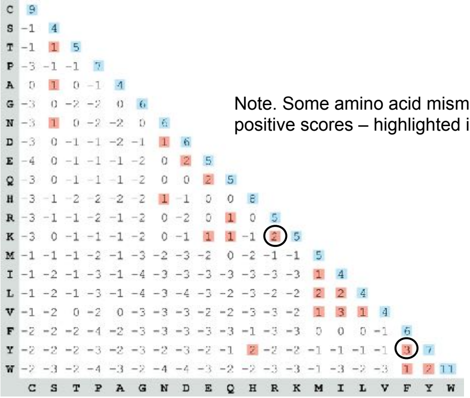
Side note: Scoring matrices

- A substitution matrix contains values proportional to the probability that amino acid *i* mutates into amino acid *j* for all pairs of amino acids
- Substitution matrices are constructed by assembling a large and diverse sample of verified pairwise alignments (or multiple sequence alignments) of amino acids.
- Substitution matrices should reflect the probabilities of mutations occurring through a period of evolution
- The two major types of substitution matrices are **PAM** and **BLOSUM**

BLOSUM62 is the default BLASTp scoring matrix

- BLOSUM matrices are based on short, ungapped blocks of conserved amino acid sequences from multiple alignments
  - – members of a block that have a most X percent sequence identity to each other are used to generate a BLOSUM**X** matrix
  - – For example, using a cutoff of 62% identity will generate the BLOSUM62 matrix
- PAM matrices are similar but built from multiple alignments where amino acid substitutions are at rate of 1% (PAM 1)
  - – Matrix multiplication is used generate higher PAM matrices
  - – PAM3 = (PAM1 x PAM1 x PAM1) etc...

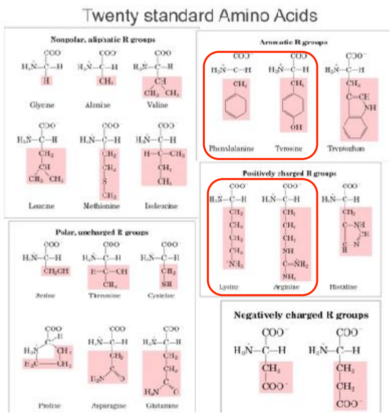
By default BLASTp Match scores come from the BLOSUM62 matrix



C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Note. Some amino acid mismatches have positive scores – highlighted in red

Protein scoring matrices reflect the properties of amino acids



Two problems standard BLAST cannot solve

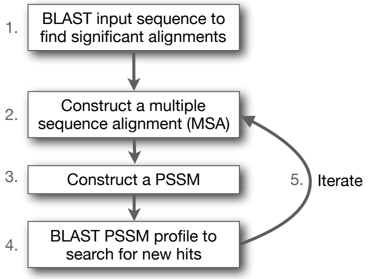
- Use human beta globin as a query against human RefSeq proteins, and blastp does not “find” human myoglobin
  - This is because the two proteins are too distantly related
  - **PSI-BLAST** at NCBI as well as hidden Markov models (HMMs) easily solve this problem
- How can we search using 10,000 base pairs as a query, or even millions of base pairs?
  - Many BLAST-like tools for genomic DNA are now available such as Megablast

PSI-BLAST: Position specific iterated BLAST

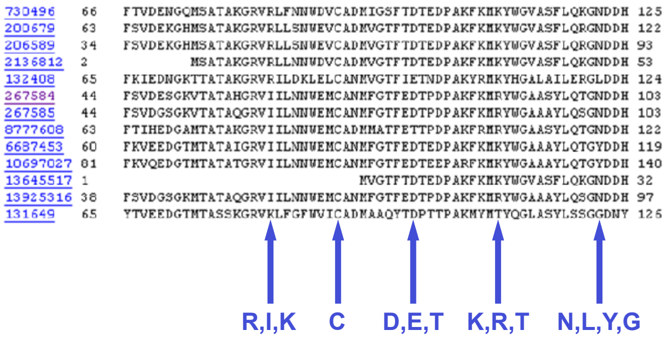
- The purpose of PSI-BLAST is to look deeper into the database for matches to your query protein sequence by employing a scoring matrix that is customized to your query
  - PSI-BLAST constructs a multiple sequence alignment from the results of a first round BLAST search and then creates a “profile” or specialized **position-specific scoring matrix** (**PSSM**) for subsequent search rounds

PSI-BLAST: Position-Specific Iterated BLAST

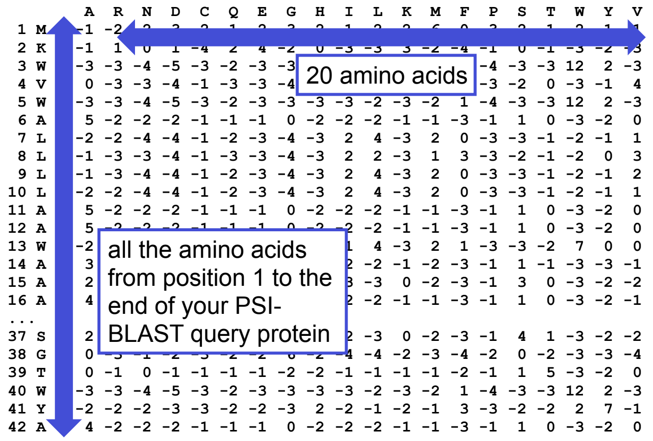
- Many proteins in a database are too distantly related to a query to be detected using standard BLAST. In many other cases matches are detected but are so distant that the inference of homology is unclear. Enter the more sensitive PSI-BLAST


1. BLAST input sequence to find significant alignments
2. Construct a multiple sequence alignment (MSA)
3. Construct a PSSM
4. BLAST PSSM profile to search for new hits
5. Iterate

Inspect the blastp output to identify empirical “rules” regarding amino acids tolerated at each position

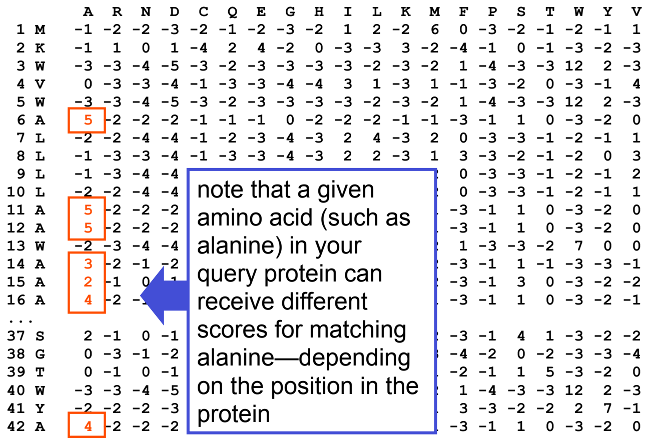


```
730496   66  FTVDENGQMSATAKGRVRLFNNWDVCADMIGSFTDTEDPAKFKMKYWGVASFLQKGNDDH 125
200679   63  FSVDEKGHMSATAKGRVRLLSNWEVCADMVGTFTDTEDPAKFKMKYWGVASFLQRGNDDH 122
206589   34  FSVDEKGHMSATAKGRVRLLSNWEVCADMVGTFTDTEDPAKFKMKYWGVASFLQRGNDDH 93
2136812  2            MSATAKGRVRLLNNWDVCADMVGTFTDTEDPAKFKMKYWGVASFLQKGNDDH 53
132408   65  FKIEDNGKTTATAKGRVRILDKLELCANMVGTFIETNDPAKYRMKYHGALAILERGLDDH 124
267584   44  FSVDESGKVTATAHGRVIILNNWEMCANMFGTFEDTPDPAKFKMRYWGAASYLQTGNDDH 103
267585   44  FSVDGSGKVTATAQGRVIILNNWEMCANMFGTFEDTPDPAKFKMRYWGAAAYLQSGNDDH 103
8777608  63  FTIHEDGAMTATAKGRVIILNNWEMCADMMATFETTPDPAKFRMRYWGAASYLQTGNDDH 122
6687453  60  FKVEEDGTMTATAIGRVIILNNWEMCANMFGTFEDTEDPAKFKMKYWGAAAYLQTGYDDH 119
10697027 81  FKVQEDGTMTATATGRVIILNNWEMCANMFGTFEDTEEPARFKMKYWGAAAYLQTGYDDH 140
13645517 1                                MVGTFTDTEDPAKFKMKYWGVASFLQKGNDDH 32
13925316 38  FSVDGSGKMTATAQGRVIILNNWEMCANMFGTFEDTPDPAKFKMRYWGAAAYLQSGNDDH 97
131649   65  YTVEEDGTMTASSKGRVKLFGFWVICADMAAQYTDPTTPAKMYMTYQGLASYLSSGGDNY 126
```



```
        A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
  1 M  -1  -2
  2 K  -1   1
  3 W  -3  -3  -4  -5  -3  -2  -3  -3                          -4  -3  -3  12   2  -3
  4 V   0  -3  -3  -4  -1  -3  -3  -4                          -3  -2   0  -3  -1   4
  5 W  -3  -3  -4  -5  -3  -2  -3  -3  -3  -3  -2  -3  -2   1  -4  -3  -3  12   2  -3
  6 A   5  -2  -2  -2  -1  -1  -1   0  -2  -2  -2  -1  -1  -3  -1   1   0  -3  -2   0
  7 L  -2  -2  -4  -4  -1  -2  -3  -4  -3   2   4  -3   2   0  -3  -3  -1  -2  -1   1
  8 L  -1  -3  -3  -4  -1  -3  -3  -4  -3   2   2  -3   1   3  -3  -2  -1  -2   0   3
  9 L  -1  -3  -4  -4  -1  -2  -3  -4  -3   2   4  -3   2   0  -3  -3  -1  -2  -1   2
 10 L  -2  -2  -4  -4  -1  -2  -3  -4  -3   2   4  -3   2   0  -3  -3  -1  -2  -1   1
 11 A   5  -2  -2  -2  -1  -1  -1   0  -2  -2  -2  -1  -1  -3  -1   1   0  -3  -2   0
 12 A   5                                  -2  -2  -1  -1  -3  -1   1   0  -3  -2   0
 13 W  -2                                       4  -3   2   1  -3  -3  -2   7   0   0
 14 A   3                                      -2  -1  -2  -3  -1   1  -1  -3  -3  -1
 15 A   2                                      -3   0  -2  -3  -1   3   0  -3  -2  -2
 16 A   4                                      -2  -1  -1  -3  -1   1   0  -3  -2  -1
...
 37 S   2                                      -3   0  -2  -3  -1   4   1  -3  -2  -2
 38 G   0                                      -4  -2  -3  -4  -2   0  -2  -3  -3  -4
 39 T   0  -1   0  -1  -1  -1  -1  -2  -2  -1  -1  -1  -1  -2  -1   1   5  -3  -2   0
 40 W  -3  -3  -4  -5  -3  -2  -3  -3  -3  -3  -2  -3  -2   1  -4  -3  -3  12   2  -3
 41 Y  -2  -2  -2  -3  -3  -2  -2  -3   2  -2  -1  -2  -1   3  -3  -2  -2   2   7  -1
 42 A   4  -2  -2  -2  -1  -1  -1   0  -2  -2  -2  -1  -1  -3  -1   1   0  -3  -2   0
```

20 amino acids
all the amino acids from position 1 to the end of your PSI-BLAST query protein



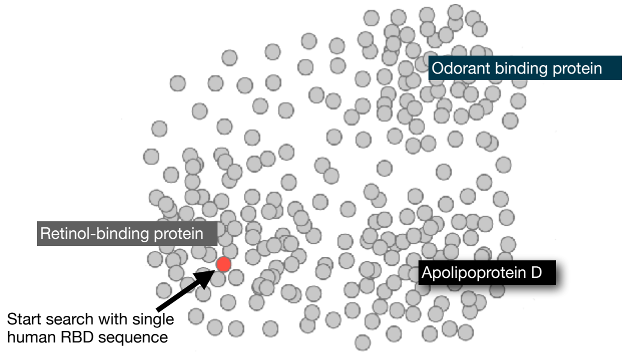
```
        A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
  1 M  -1  -2  -2  -3  -2  -1  -2  -3  -2   1   2  -2   6   0  -3  -2  -1  -2  -1   1
  2 K  -1   1   0   1  -4   2   4  -2   0  -3  -3   3  -2  -4  -1   0  -1  -3  -2  -3
  3 W  -3  -3  -4  -5  -3  -2  -3  -3  -3  -3  -2  -3  -2   1  -4  -3  -3  12   2  -3
  4 V   0  -3  -3  -4  -1  -3  -3  -4  -4   3   1  -3   1  -1  -3  -2   0  -3  -1   4
  5 W  -3  -3  -4  -5  -3  -2  -3  -3  -3  -3  -2  -3  -2   1  -4  -3  -3  12   2  -3
  6 A   5  -2  -2  -2  -1  -1  -1   0  -2  -2  -2  -1  -1  -3  -1   1   0  -3  -2   0
  7 L  -2  -2  -4  -4  -1  -2  -3  -4  -3   2   4  -3   2   0  -3  -3  -1  -2  -1   1
  8 L  -1  -3  -3  -4  -1  -3  -3  -4  -3   2   2  -3   1   3  -3  -2  -1  -2   0   3
  9 L  -1  -3  -4  -4                                       0  -3  -3  -1  -2  -1   2
 10 L  -2  -2  -4  -4                                       0  -3  -3  -1  -2  -1   1
 11 A   5  -2  -2  -2                                      -3  -1   1   0  -3  -2   0
 12 A   5  -2  -2  -2                                      -3  -1   1   0  -3  -2   0
 13 W  -2  -3  -4  -4                                       1  -3  -3  -2   7   0   0
 14 A   3  -2  -1  -2                                      -3  -1   1  -1  -3  -3  -1
 15 A   2  -1   0                                          -3  -1   3   0  -3  -2  -2
 16 A   4  -2                                              -3  -1   1   0  -3  -2  -1
...
 37 S   2  -1   0  -1                                      -3  -1   4   1  -3  -2  -2
 38 G   0  -3  -1  -2                                      -4  -2   0  -2  -3  -3  -4
 39 T   0  -1   0  -1                                      -2  -1   1   5  -3  -2   0
 40 W  -3  -3  -4  -5                                       1  -4  -3  -3  12   2  -3
 41 Y  -2  -2  -2  -3                                       3  -3  -2  -2   2   7  -1
 42 A   4  -2  -2  -2                                      -3  -1   1   0  -3  -2   0
```

note that a given amino acid (such as alanine) in your query protein can receive different scores for matching alanine—depending on the position in the protein

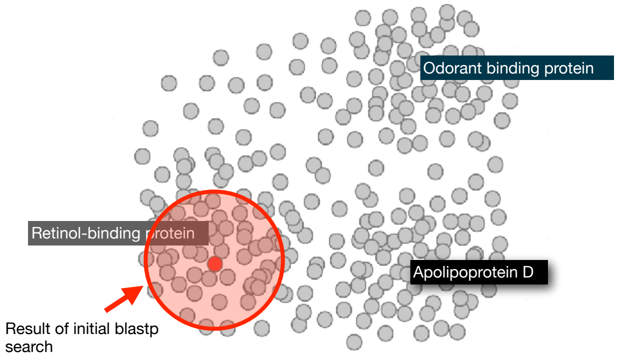
```
        A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V
  1 M  -1 -2 -2 -3 -2 -1 -2 -3 -2  1  2 -2  6  0 -3 -2 -1 -2 -1  1
  2 K                                                             -3
  3 W                                                             -3
  4 V                                                              4
  5 W                                                             -3
  6 A   5 -2 -2 -2 -1 -1 -1  0 -2 -2 -2 -1 -1 -3 -1  1  0 -3 -2  0
  7 L  -2 -2 -4 -4 -1 -2 -3 -4 -3  2  4 -3  2  0 -3 -3 -1 -2 -1  1
  8 L  -1 -3 -3 -4 -1 -3 -3 -4 -3  2  2 -3  1  3 -3 -2 -1 -2  0  3
  9 L  -1 -3 -4 -4                            0 -3 -3 -1 -2 -1  2
 10 L  -2 -2 -4 -4                            0 -3 -3 -1 -2 -1  1
 11 A   5 -2 -2 -2                           -3 -1  1  0 -3 -2  0
 12 A   5 -2 -2 -2                           -3 -1  1  0 -3 -2  0
 13 W  -2 -3 -4 -4                            1 -3 -3 -2  7  0  0
 14 A   3 -2 -1 -2                           -3 -1  1 -1 -3 -3 -1
 15 A   2 -1  0                              -3 -1  3  0 -3 -2 -2
 16 A   4 -2 -1                              -3 -1  1  0 -3 -2 -1
...
 37 S   2 -1  0 -1                           -3 -1  4  1 -3 -2 -2
 38 G   0 -3 -1 -2                           -4 -2  0 -2 -3 -3 -4
 39 T   0 -1  0 -1                           -2 -1  1  5 -3 -2  0
 40 W  -3 -3 -4 -5                            1 -4 -3 -3 12  2 -3
 41 Y  -2 -2 -2 -3                            3 -3 -2 -2  2  7 -1
 42 A   4 -2 -2 -2                           -3 -1  1  0 -3 -2  0
```

The PSI-BLAST PSSM is essentially a query customized scoring matrix that is more sensitive than PAM or BLOSUM.

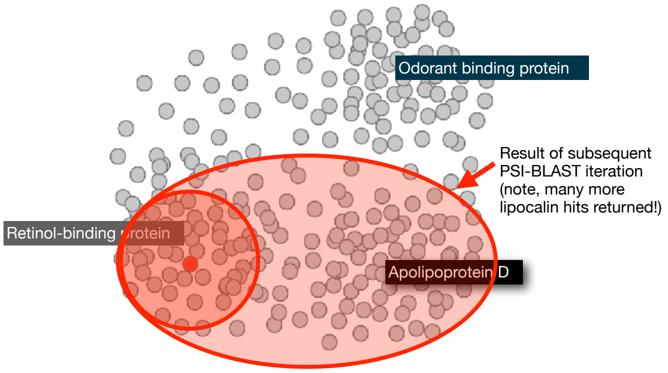
note that a given amino acid (such as alanine) in your query protein can receive different scores for matching alanine—depending on the position in the protein



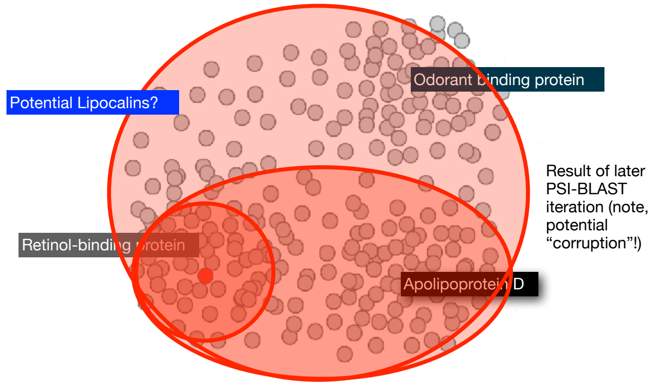
Start search with single human RBD sequence



Result of initial blastp search



Result of subsequent PSI-BLAST iteration (note, many more lipocalin hits returned!)



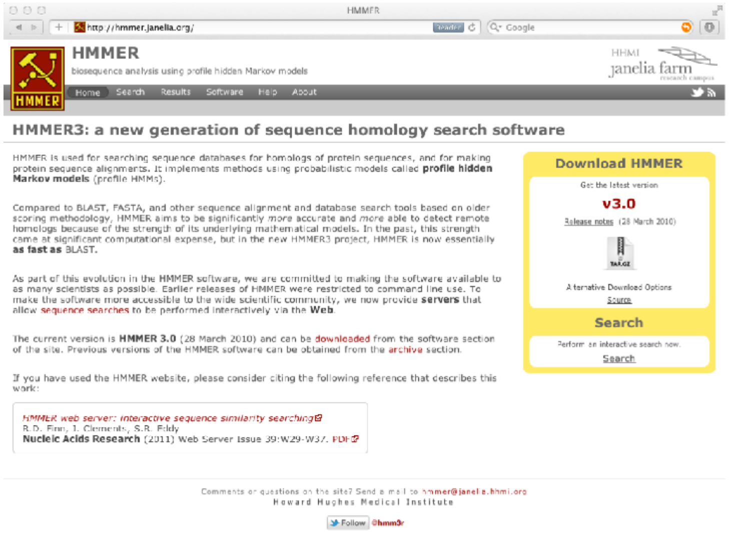
Result of later PSI-BLAST iteration (note, potential "corruption"!)

PSI-BLAST returns dramatically more hits

- The search process is continued iteratively, typically about five times, and at each step a new PSSM is built
  - You must decide how many iterations to perform and which sequences to include!
  - You can stop the search process at any point - typically whenever few new results are returned or when no new "sensible" results are found

Iteration	Hits with E < 0.005	Hits with E > 0.005
1	34	61
2	314	79
3	416	57
4	432	50
5	432	50

Human retinol-binding protein 4 (RBP4; P02753) was used as a query in a PSI-BLAST search of the RefSeq database.


HMMER
http://hmmer.janelia.org/
Google
HMMER
biosequence analysis using profile hidden Markov models
HHMI janelia farm research campus
Home Search Results Software Help About

HMMER3: a new generation of sequence homology search software

HMMER is used for searching sequence databases for homologs of protein sequences, and for making protein sequence alignments. It implements methods using probabilistic models called **profile hidden Markov models** (profile HMMs).

Compared to BLAST, FASTA, and other sequence alignment and database search tools based on older scoring methodology, HMMER aims to be significantly *more* accurate and *more* able to detect remote homologs because of the strength of its underlying mathematical models. In the past, this strength came at significant computational expense, but in the new HMMER3 project, HMMER is now essentially **as fast as** BLAST.

As part of this evolution in the HMMER software, we are committed to making the software available to as many scientists as possible. Earlier releases of HMMER were restricted to command line use. To make the software more accessible to the wide scientific community, we now provide **servers** that allow sequence searches to be performed interactively via the **Web**.

The current version is **HMMER 3.0** (28 March 2010) and can be downloaded from the software section of the site. Previous versions of the HMMER software can be obtained from the archive section.

If you have used the HMMER website, please consider citing the following reference that describes this work:

HMMER web server: interactive sequence similarity searching
R.D. Finn, J. Clements, S.R. Eddy
Nucleic Acids Research (2011) Web Server Issue 39:W29-W37. PDF

Download HMMER

Get the latest version

v3.0

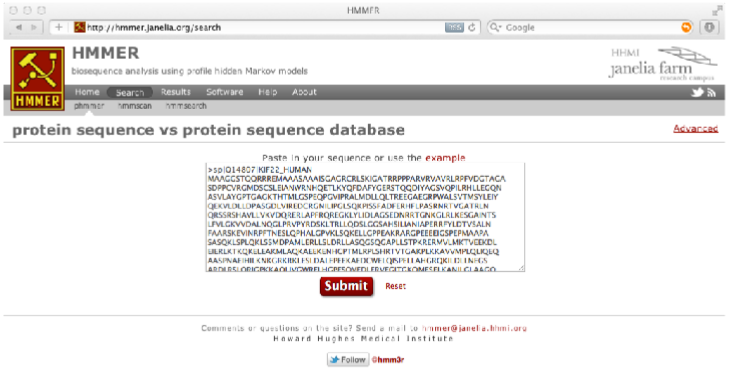
Release notes (28 March 2010)

Alternative Download Options
Source

Search

Perform an interactive search now.
Search

Comments or questions on the site? Send a mail to hmmer@janelia.hhmi.org
Howard Hughes Medical Institute


HMMER
http://hmmer.janelia.org/search
Google
HMMER
biosequence analysis using profile hidden Markov models
HHMI janelia farm research campus
Home Search Results Software Help About
phmmer hmmscan hmmsearch

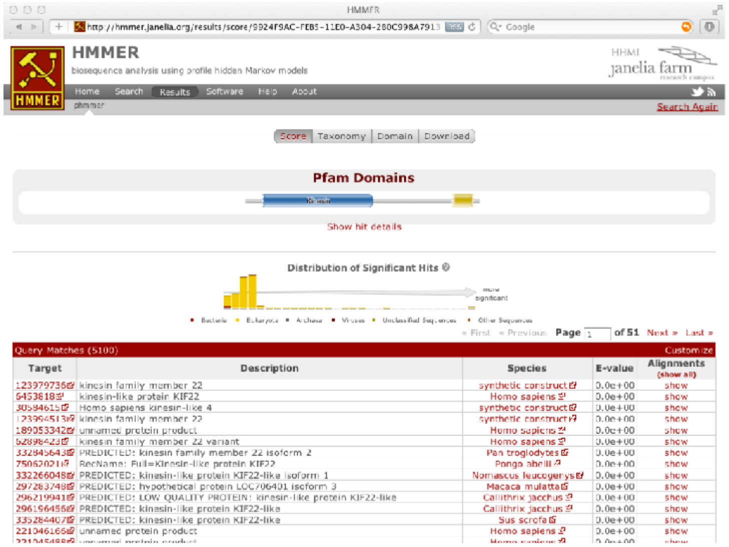
protein sequence vs protein sequence database

Advanced

Paste in your sequence or use the example

```
>sp|O14807|KIF22_HUMAN
MAAGGSTQQRRREMAAASAAAISGAGRCRLSKIGATRRPPPARVRVAVRLRPFVDGTACA
SDPPCVRGMDSCSLEIANWRNHQETLKYQFDAFYGERSTQQDIYAGSVQPILRHLLEGQN
ASVLAYGPTGAGKTHTMLGSPEQPGVIPRALMDLLQLTREEGAEGRPWALSVTMSYLEIY
QEKVLDLLDPASGDLVIREDCRGNILIPGLSQKPISSFADFERHFLPASRNRTVGATRLN
QRSSRSHAVLLVKVDQRERLAPFRQREGKLYLIDLAGSEDNRRTGNKGLRLKESGAINTS
LFVLGKVVDALNQGLPRVPYRDSKLTRLLQDSLGGSAHSILIANIAPERRFYLDTVSALN
FAARSKEVINRPFTNESLQPHALGPVKLSQKELLGPPEAKRARGPEEEEIGSPEPMAAPA
SASQKLSPLQKLSSMDPAMLERLLSLDRLLASQGSQGAPLLSTPKRERMVLMKTVEEKDL
EIERLKTKQKELEAKMLAQKAEEKENHCPTMLRPLSHRTVTGAKPLKKAVVMPLQLIQEQ
AASPNAEIHILKNKGRKRKLESLDALEPEEKAEDCWELQISPELLAHGRQKILDLLNEGS
ARDLRSLQRIGPKKAQLIVGWRELHGPFSQVEDLERVEGITGKQMESFLKANILGLAAGQ
```

Comments or questions on the site? Send a mail to hmmer@janelia.hhmi.org
Howard Hughes Medical Institute


HMMER
http://hmmer.janelia.org/results/score/9924F9AC-FEB5-11E0-A304-2B0C998A7913
Google
HMMER
biosequence analysis using profile hidden Markov models
HHMI janelia farm research campus
Home Search Results Software Help About
phmmer
Search Again
Score Taxonomy Domain Download

Pfam Domains

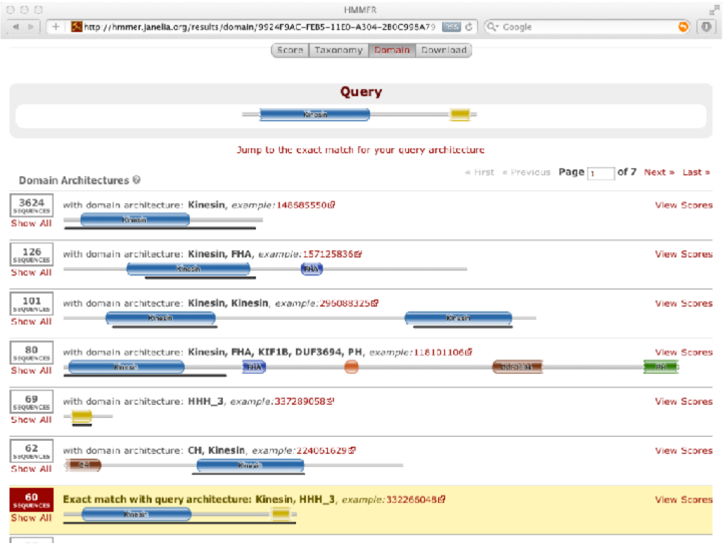
Distribution of Significant Hits

more significant
Bacteria
Eukaryota
Archaea
Viruses
Unclassified Sequences
Other Sequences

« First « Previous Page 1 of 51 Next » Last »

Query Matches (5100) Customize

Target	Description	Species	E-value	Alignments (show all)
123979736	kinesin family member 22	synthetic construct	0.0e+00	show
6453818	kinesin-like protein KIF22	Homo sapiens	0.0e+00	show
30584615	Homo sapiens kinesin-like 4	synthetic construct	0.0e+00	show
123994513	kinesin family member 22	synthetic construct	0.0e+00	show
189053342	unnamed protein product	Homo sapiens	0.0e+00	show
62898423	kinesin family member 22 variant	Homo sapiens	0.0e+00	show
332845643	PREDICTED: kinesin family member 22 isoform 2	Pan troglodytes	0.0e+00	show
75062021	RecName: Full=Kinesin-like protein KIF22	Pongo abelii	0.0e+00	show
332266048	PREDICTED: kinesin-like protein KIF22-like isoform 1	Nomascus leucogenys	0.0e+00	show
297283748	PREDICTED: hypothetical protein LOC706401 isoform 3	Macaca mulatta	0.0e+00	show
296219941	PREDICTED: LOW QUALITY PROTEIN: kinesin-like protein KIF22-like	Callithrix jacchus	0.0e+00	show
296196456	PREDICTED: kinesin-like protein KIF22-like	Callithrix jacchus	0.0e+00	show
335284407	PREDICTED: kinesin-like protein KIF22-like	Sus scrofa	0.0e+00	show
221046166	unnamed protein product	Homo sapiens	0.0e+00	show
221045488	unnamed protein product	Homo sapiens	0.0e+00	show


HMMER
http://hmmer.janelia.org/results/domain/9924F9AC-FEB5-11E0-A304-2B0C998A79
Google
Score Taxonomy Domain Download

Query

Jump to the exact match for your query architecture

Domain Architectures

« First « Previous Page 1 of 7 Next » Last »

3624 SEQUENCES Show All — with domain architecture: **Kinesin**, *example:* 148685550 — View Scores

126 SEQUENCES Show All — with domain architecture: **Kinesin, FHA**, *example:* 157125836 — View Scores

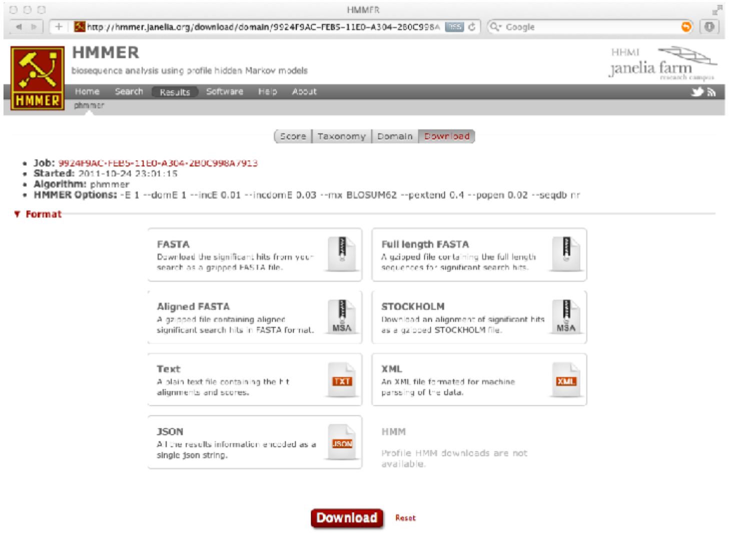
101 SEQUENCES Show All — with domain architecture: **Kinesin, Kinesin**, *example:* 296088325 — View Scores

80 SEQUENCES Show All — with domain architecture: **Kinesin, FHA, KIF1B, DUF3694, PH**, *example:* 118101106 — View Scores

69 SEQUENCES Show All — with domain architecture: **HHH_3**, *example:* 337289058 — View Scores

62 SEQUENCES Show All — with domain architecture: **CH, Kinesin**, *example:* 224051629 — View Scores

60 SEQUENCES Show All — **Exact match with query architecture: Kinesin, HHH_3**, *example:* 332266048 — View Scores


HMMER
http://hmmer.janelia.org/download/domain/9924F9AC-FEB5-11E0-A304-2B0C998A
Google
HMMER
biosequence analysis using profile hidden Markov models
HHMI janelia farm research campus
Home Search Results Software Help About
phmmer
Score Taxonomy Domain Download

- **Job:** 9924F9AC-FEB5-11E0-A304-2B0C998A7913
- **Started:** 2011-10-24 23:01:15
- **Algorithm:** phmmer
- **HMMER Options:** -E 1 --domE 1 --incE 0.01 --incdomE 0.03 --mx BLOSUM62 --pextend 0.4 --popen 0.02 --seqdb nr

▼ **Format**

FASTA
Download the significant hits from your search as a gzipped FASTA file.

Full length FASTA
A gzipped file containing the full length sequences for significant search hits.

Aligned FASTA
A gzipped file containing aligned significant search hits in FASTA format.

STOCKHOLM
Download an alignment of significant hits as a gzipped STOCKHOLM file.

Text
A plain text file containing the hit alignments and scores.

XML
An XML file formated for machine parsing of the data.

JSON
All the results information encoded as a single json string.

HMM
Profile HMM downloads are not available.

Summary

- Alignment basics
  - Why compare biological sequences?
- Homologue detection
  - Orthologs, paralogs, similarity and identity
  - Sequence changes during evolution
  - Alignment view: matches, mismatches and gaps
- Pairwise sequence alignment methods
  - Brute force alignment
  - Dot matrices
  - Dynamic programing

  (global vs local alignment)
- Rapid heuristic approaches
  - BLAST
- Practical database searching
  - BLAST, PSI-BLAST and HMM approaches