PAIRWISE SEQUENCE ALIGNMENT AND DATABASE SEARCHING

Barry Grant
University of Michigan
www.thegrantlab.org

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

- NCBI BLAST frustrations
- Need for FASTA header lines ">example1"
- More on protein structure viewing and finding
- "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

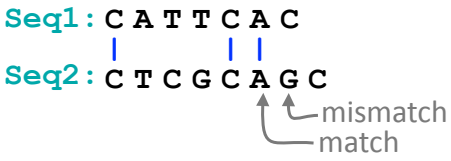
TODAYS MENU

- 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.

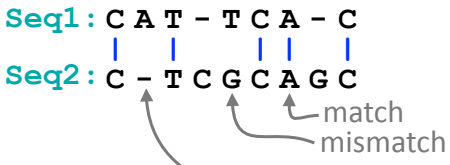
```
Seq1: C A T T C A C
Seq2: C T C G C A G C
```

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

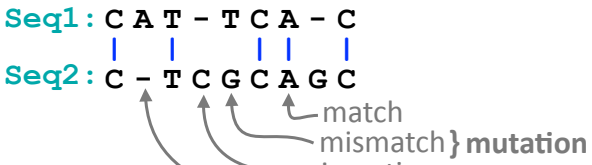
Basic Idea: Display one sequence above another with spaces (termed **gaps**) inserted in both to reveal **similarity** of nucleotides or amino acids.



Add gaps to increase number of matches

gaps

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 prediction
- **Assembly of sequen** construct such as a bac
- **Ma** **s to a known genome**
  - looking for differences from reference
  - SNPs, 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!

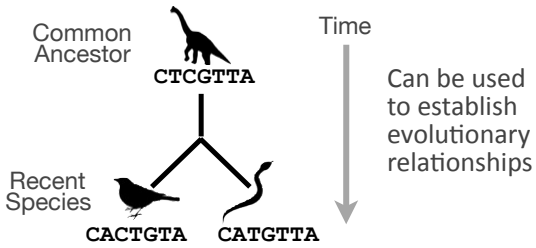
Outline for today

- 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



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:

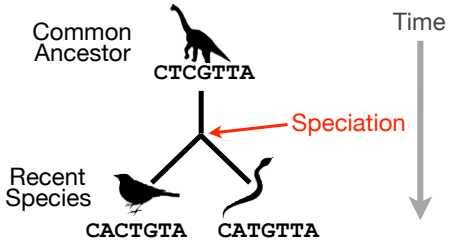
– **Identity** and/or **Similarity**

(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.

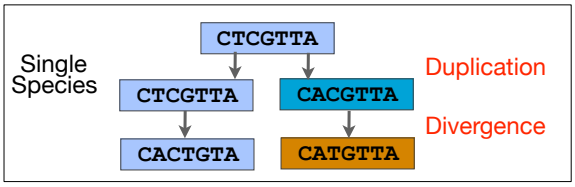
– Ortho = [greek: straight] ... implies direct descent



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]



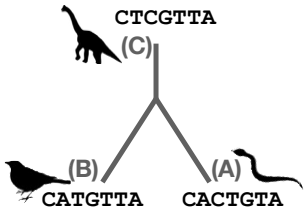
Orthologs *vs* Paralogs

- 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.

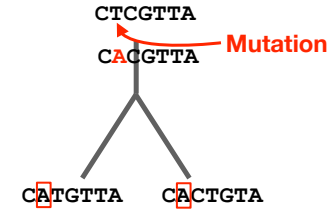
- Mutations/Substitutions
- Deletions
- Insertions



Mutations, deletions and insertions

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

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

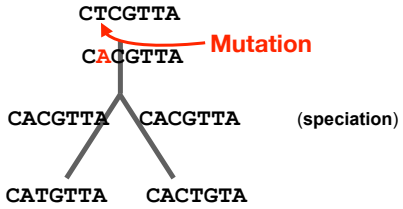


Likely occurred prior to speciation

Mutations, deletions and insertions

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

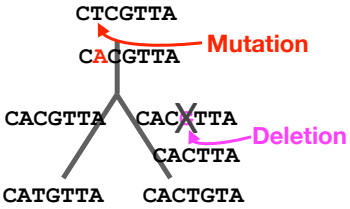
- Mutations/Substitutions CTCGTTA → CACGTTA
- Deletions
- Insertions



Mutations, deletions and insertions

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

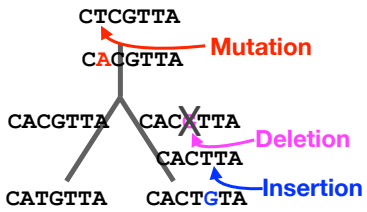
- Mutations/Substitutions CTCGTTA → CACGTTA
- **Deletions** CACGTTA → CACTTA
- Insertions



Mutations, deletions and insertions

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

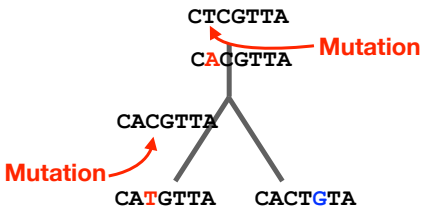
- Mutations/Substitutions CTCGTTA → CACGTTA
- Deletions CACGTTA → CACTTA
- **Insertions** CACTTA → CACTGTA



Mutations, deletions and insertions

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

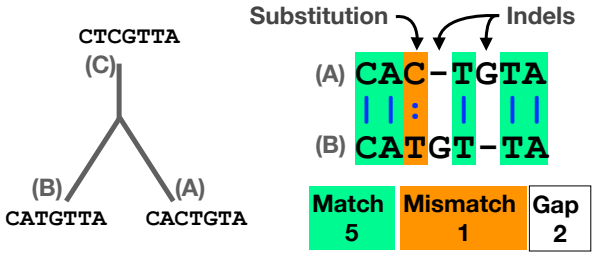
- **Mutations/Substitutions** CTCGTTA → CACGTTA
- Deletions CACGTTA → CATGTTA
- Insertions



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)

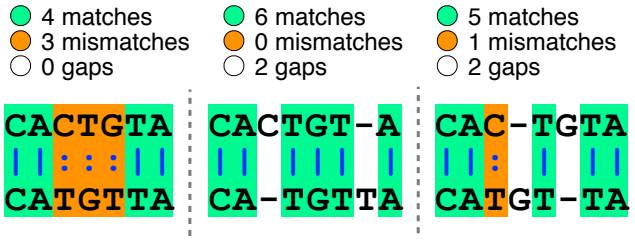


Alternative alignments

- 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?

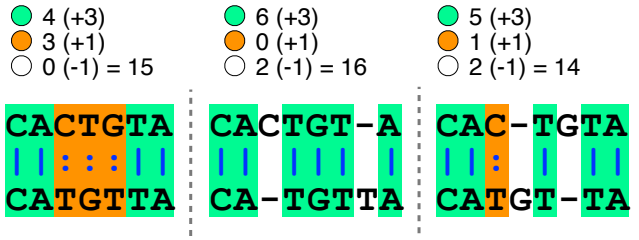
Alternative alignments

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



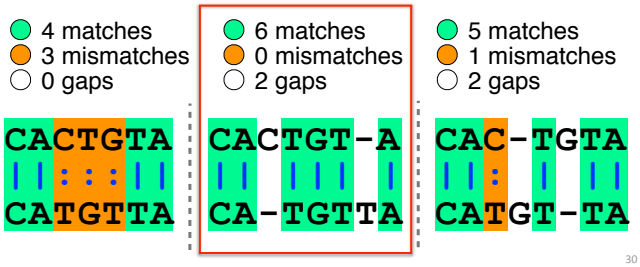
Scoring alignments

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



Optimal alignments

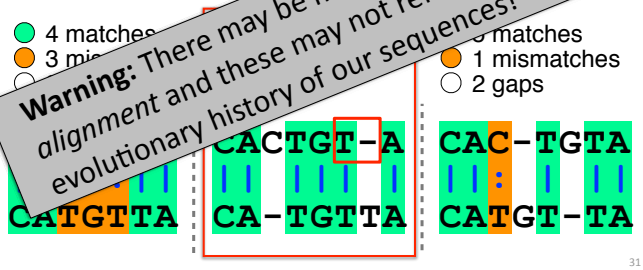
- Biologists often prefer **parsimonious alignments**, where the number of postulated sequence changes is minimized.



Optimal alignments

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

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



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?

- 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)

```
GTAATCTG
        TTAAGCTGA

GTAATCTG
      | |
      TTAAGCTGA

GTAATCTG
      |
      TTAAGCTGA

GTAATCTG
      | | | | |
      TTAAGCTGA

GTAATCTG
      |
      TTAAGCTGA

GTAATCTG
      ||
      TTAAGCTGA

GTAATCTG
        TTAAGCTGA

GTAATCTG
        |
      TTAAGCTGA

GTAATCTG
      | |
      TTAAGCTGA

GTAATCTG
        TTAAGCTGA

GTAATCTG
        TTAAGCTGA
```

Brute Force Alignment, No Gaps

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:

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

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

Slide from Jeffery de Wet

Three general solutions to the alignment problem

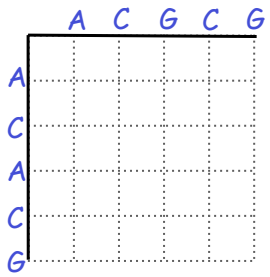
- The **dot plot** or **dot matrix** approach
  - A simple graphical method for pair-wise alignment
  - No scoring, so difficult to compare alternative alignments
  - Can give visual clues to sequence structure but requires human interaction
- **Dynamic programming** algorithms
  - Provides Optimal solutions (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

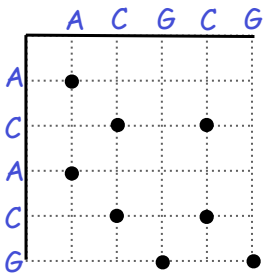
Dot plots: simple graphical approach

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



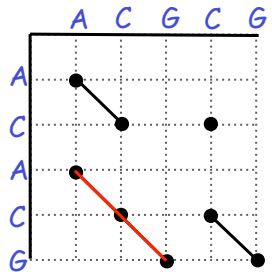
Dot plots: simple graphical approach

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



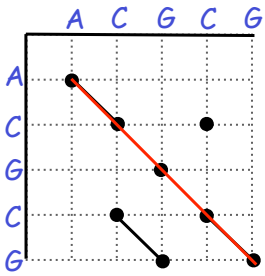
Dot plots: simple graphical approach

- Diagonal runs of dots indicate matched segments of sequence



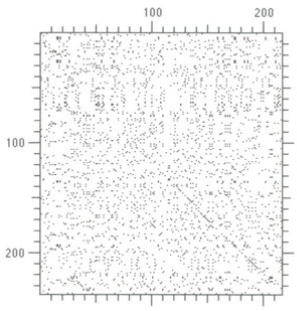
Dot plots: simple graphical approach

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



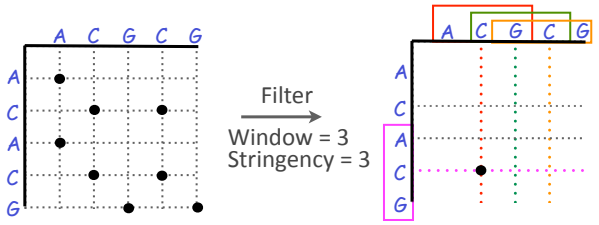
Dot plots: simple graphical approach

- Dot matrices for long sequences can be noisy



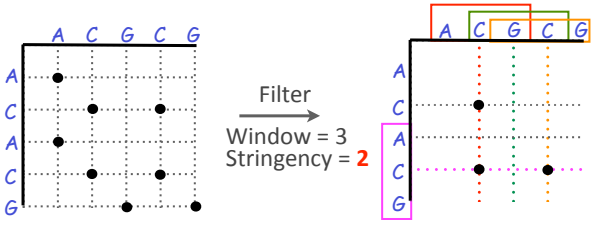
Dot plots: window size and match stringency

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

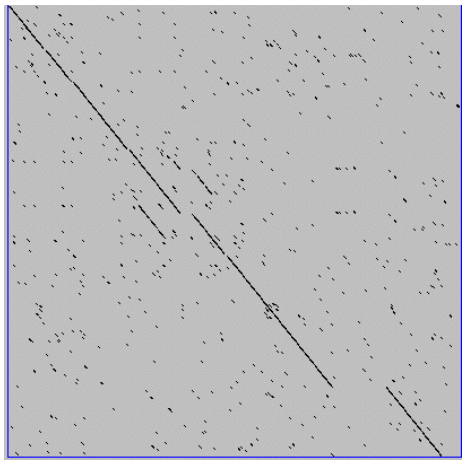


Dot plots: window size and match stringency

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



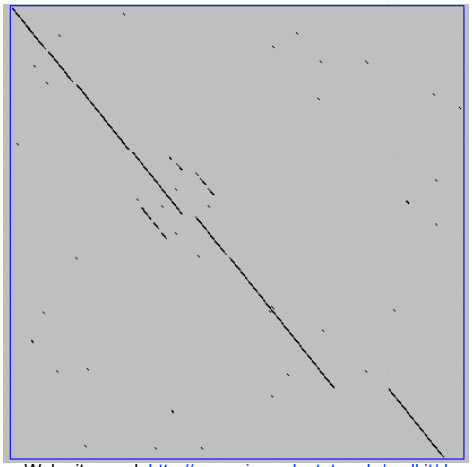
Window size = 5 bases



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



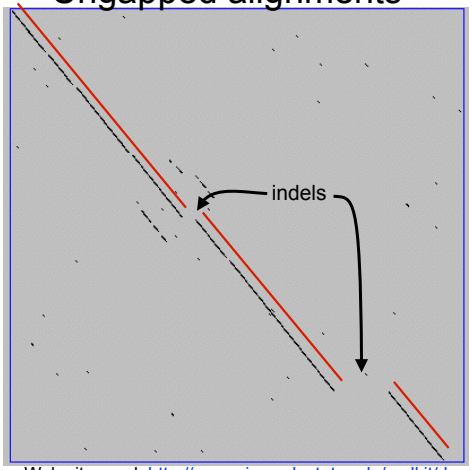
This is a dot plot of the same sequence pair. Now 7 bases in a row must match for a dot to be place. Noise is reduced.

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

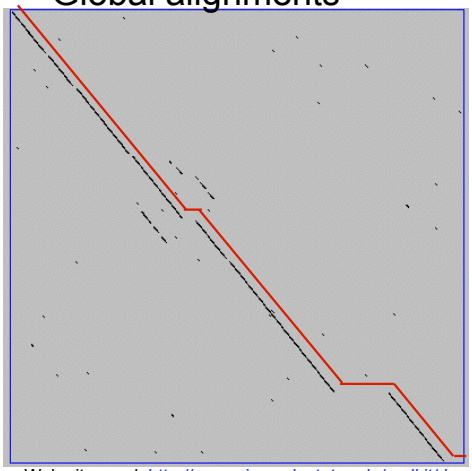


Only **diagonals** can be followed.

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.

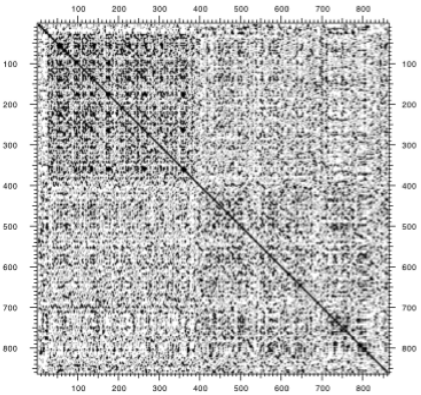
Global alignments do not have good statistical characterization and are **not used for database searches.**

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

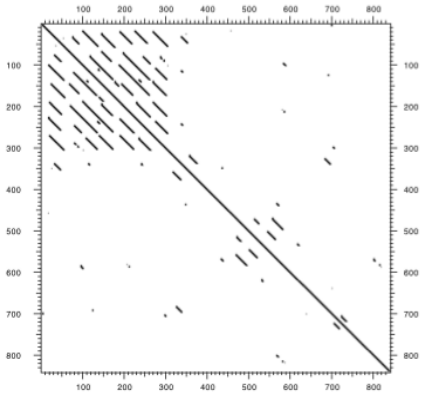


Human LDL receptor protein sequence (Genbank P01130)

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"

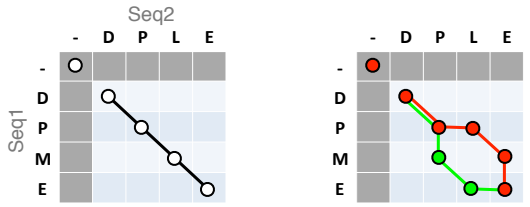
- Some matches can be rewarded more than others, depending on likelihood
- Use PAM or BLOSUM **substitution matrix**
  - (more on these later)
- Put a dot only if a minimum total or average weight is achieved
  - See chapter 3 in *Mount, "Bioinformatics sequence and genome analysis"*.

Three general solutions to the alignment problem

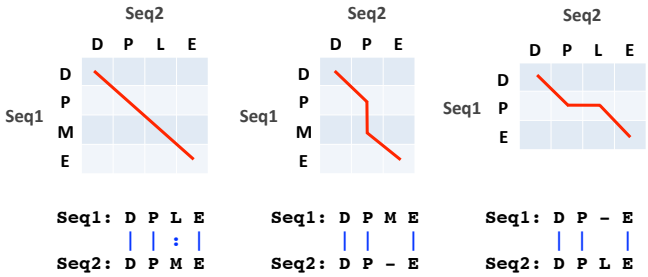
- The **dot plot** or **dot matrix** approach
  - A simple graphical method for pair-wise alignment
  - No scoring, so difficult to compare alternative alignments
  - Can give visual clues to sequence structure but requires human interaction
- **Dynamic programming** algorithms
  - Provides Optimal solutions (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**



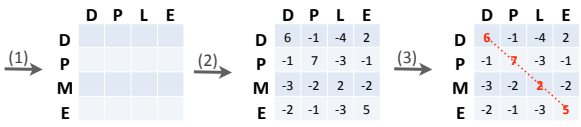
Different paths represent different alignments



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

Algorithm of **Needleman and Wunsch**

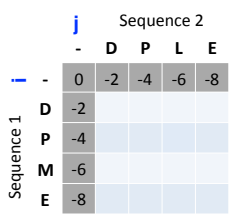
- The Needleman–Wunsch approach to global sequence alignment has three basic steps:
  (1) setting up a 2D-grid (or **alignment matrix**),
  (2) **scoring the matrix**, and
  (3) identifying the **optimal path** through the matrix



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

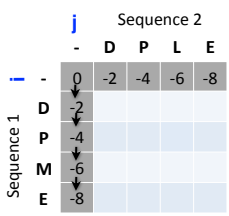
- Start by filling in the first row and column – these are all indels (gaps).
  - Each step you take you will add the **gap penalty** to the score ($S_{i,j}$) accumulated in the previous cell



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

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



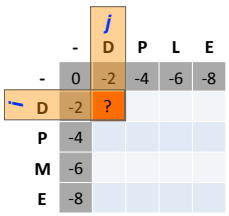
Scores: match = +1, mismatch = -1, gap = -2

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

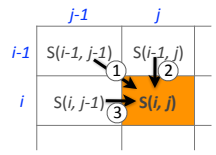
Seq1: DPME
Seq2: ----

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

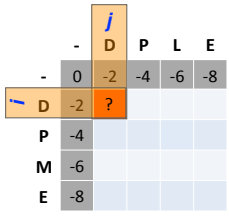


Scores: match = +1, mismatch = -1, gap = -2



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

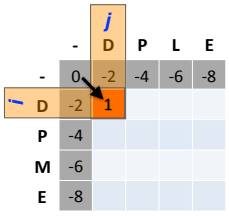


Scores: match = +1, mismatch = -1, gap = -2

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

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



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

① (0)+(+1) = +1 <= (D-D) match!

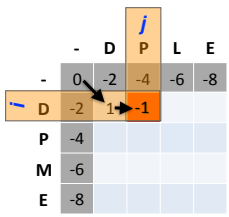
② (-2)+(-2) = -4

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)



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

① (-2)+(-1) = -3 <= (D-P) mismatch!

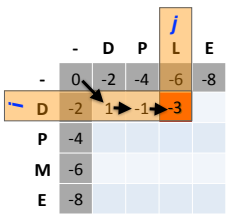
② (-4)+(-2) = -6

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.



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

① (-4)+(-1) = -5 <= (D-L) mismatch

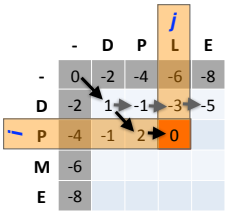
② (-6)+(-2) = -8

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.



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

① (-1)+(-1) = -2

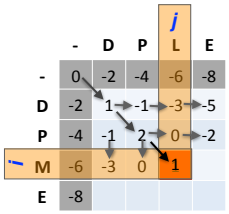
② (-3)+(-2) = -5

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



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

① (2)+(**-1**) = 0 <= mismatch

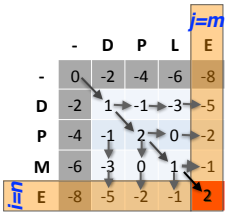
② (0)+(-2) = -2

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)



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

① (+1)+(+1) = +2

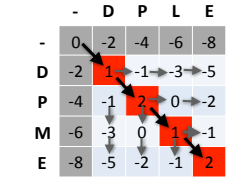
② (-1)+(-2) = -3

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



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

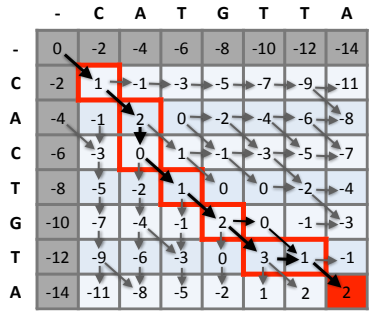
Questions:

- 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

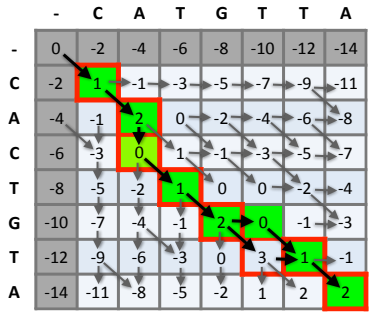
Questions:

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



More than one alignment possible

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



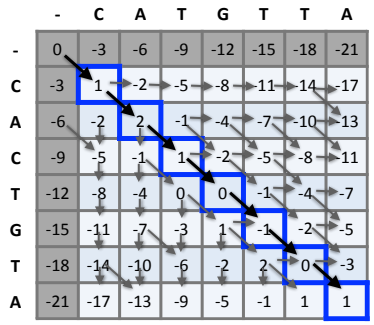
Alignment

```
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



Alignment

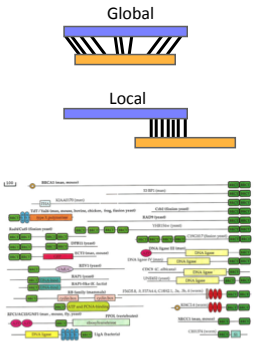
```
CACTGT-A
CA-TGTTA

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

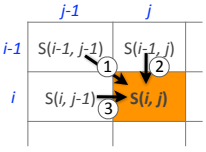
- Smith & Waterman proposed simply that a local alignment of two sequences allow arbitrary-length segments of each sequence to be aligned, with no penalty for the unaligned portions of the sequences. Otherwise, the score for a local alignment is calculated the same way as that for a global alignment

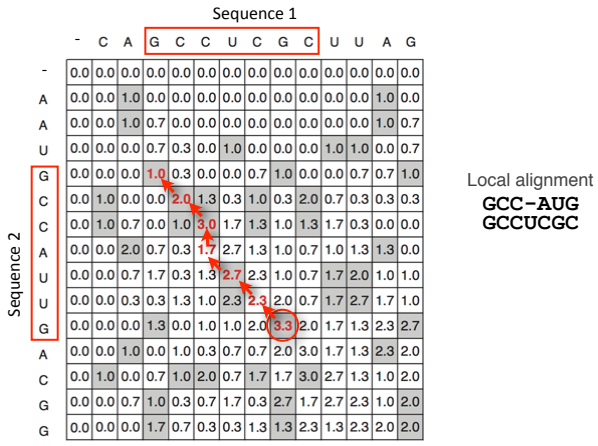
Smith, T.F. & Waterman, M.S. (1981) "Identification of common molecular subsequences." J. Mol. Biol. 147:195-197.

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

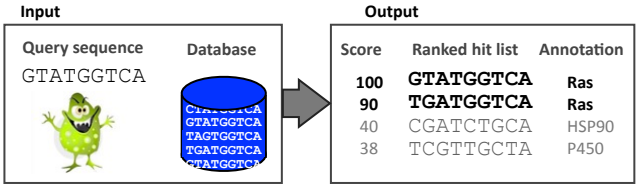
$$S(i,j) = \text{Max}\begin{cases} S(i-1,j-1) + (\text{mis})\text{match} & \searrow\ (1) \\ S(i-1,j) - \text{gap penalty} & \downarrow\ (2) \\ S(i,j-1) - \text{gap penalty} & \rightarrow\ (3) \\ 0 & (4) \end{cases}$$





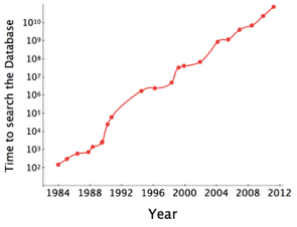
Local alignments can be used for database searching

- **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



The database search problem

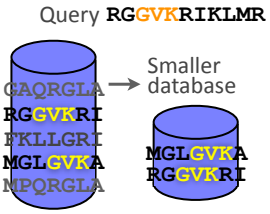
- Due to the rapid growth of sequence databases, search algorithms have to be both efficient and sensitive
  - Time to search with SW is proportional to *m* x *n* (*m* is length of query, n is length of database), **too slow for large databases**!



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**!



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**

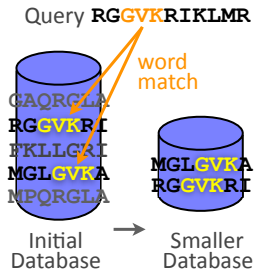
- BLAST (Basic Local Alignment Search Tool) is a simplified form of Smith-Waterman (SW) alignment that is popular because it is **fast** and **easily accessible**
  - BLAST is a heuristic approximation to SW - It examines only part of the search space
  - BLAST saves time by restricting the search by 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

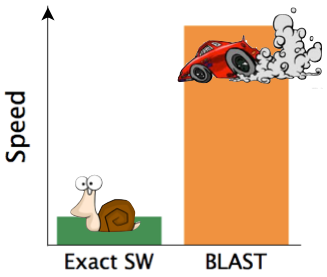
Rapid, heuristic versions of Smith–Waterman: **BLAST**

- BLAST (Basic Local Alignment Search Tool) i… ied form of Smith-Waterman (SW) alig… ar because it is **fast** and **easily**
  - BLAST finds regions… sequences
  - BL… …he search by scanning …ikely matches before performing …ignments
  - …some sensitivity in exchange for speed
  - In 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
  GVK
   VKR
    KRI
```

generate list of **w=3 words** for query

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 ...
  GVK GAK GIK GGK ...
   VKR VRR VHR VER ...
    KRI KKI KHI KDI ...
```

extend list of words similar to query

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 ...
    GGV GAV GTV GCV ...
     GVK GLK GIK GGK ...
      VKR VRR VHR VER ...
       KRI KKI KHI KDI ...
```

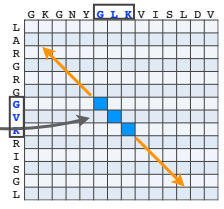
search for **perfect matches** in the database sequence

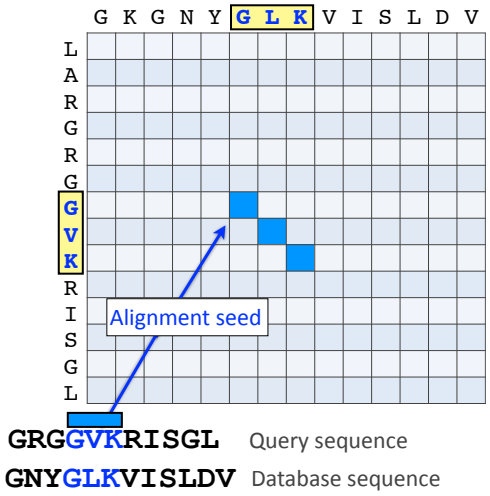
Blast

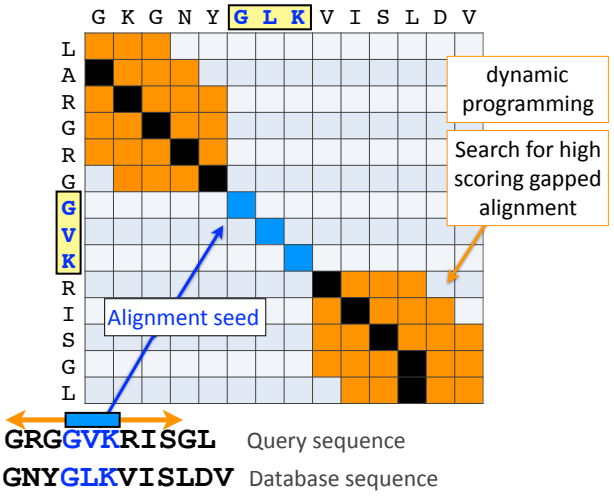
- **Phase 4**: the initial database hits are extended in both directions using dynamic programing

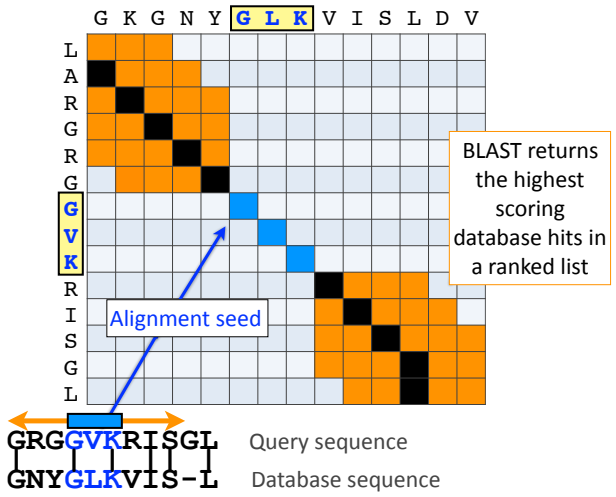
```
GNYGLKVISLDVE   Database sequence
  RGGVKRI       Query sequence
```

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









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	Total score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo	677	677	100%	0	100%	NP_004512.1
Kif5b protein [Mus musculus]	676	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	48.2	52	40%	0.03	32%	ELK35081.1
mKIAA4102 protein [Mus musculus]	42.7	42.7	38%	3.02	24%	EHH28205.1

Statistical significance of results

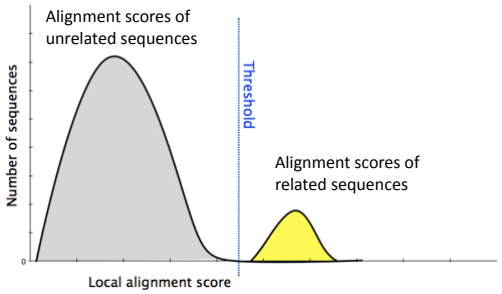
- An important feature of BLAST is the computation of statistical significance for each hit. This is described by the **E value** (expect value)

Description	Max score	Total score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo	677	677	100%	0	100%	NP_004512.1
Kif5b protein [Mus musculus]	676	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	48.2	52	40%	0.03	32%	ELK35081.1
mKIAA4102 protein [Mus musculus]	42.7	42.7	38%	3.02	24%	EHH28205.1

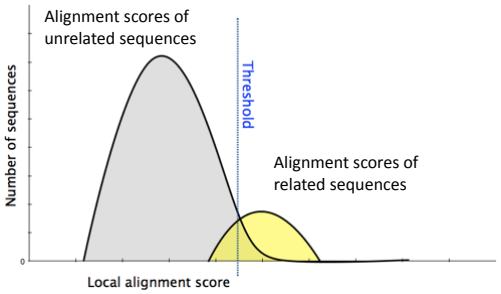
BLAST scores and E-values

- 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

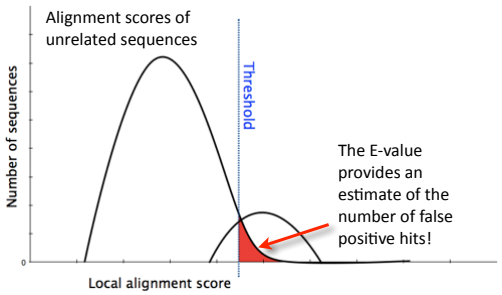
- Ideally, a threshold separates all query related sequences (yellow) from all unrelated sequences (gray)



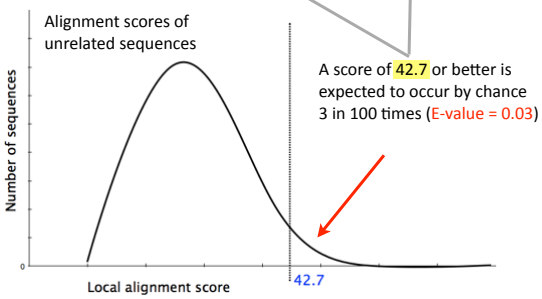
- Unfortunately, often both score distributions overlap
  - The E value describes the expected number of hits with a score above the threshold if the query and database are unrelated



- 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



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
Kif5b protein [Mus musculus]	676	676	100%	0	98%	AAA20133.1
Kinesin-14 heavy chain [Danio rerio]	595	595	88%	0	78%	XP_00320703
hypothetical protein EGK_18589	42.7	52	40%	0.03	32%	ELK35081.1

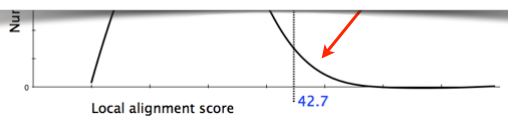


Description	Max score	Total score	Query cover	E value	Max ident	Accession
kinesin-1 heavy chain [Homo	677	677	100%	0	100%	NP_004512.1

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:

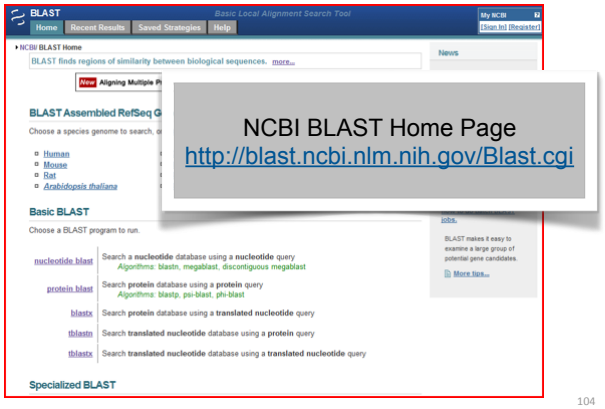
http://www.ncbi.nlm.nih.gov/blast/tutorial/Altschul-1.html



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



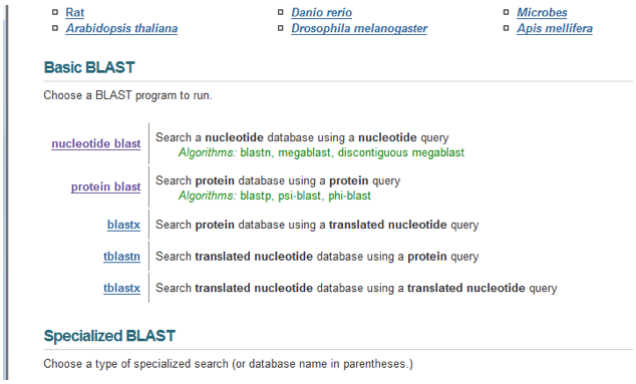
Practical database searching with BLAST

- 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"

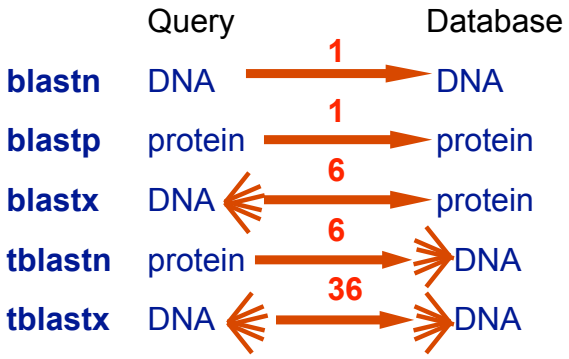
Step 1: Choose your sequence

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

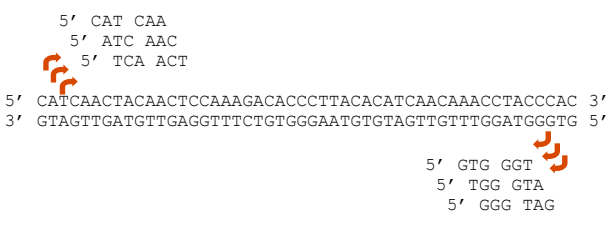
Step 2: Choose the BLAST program

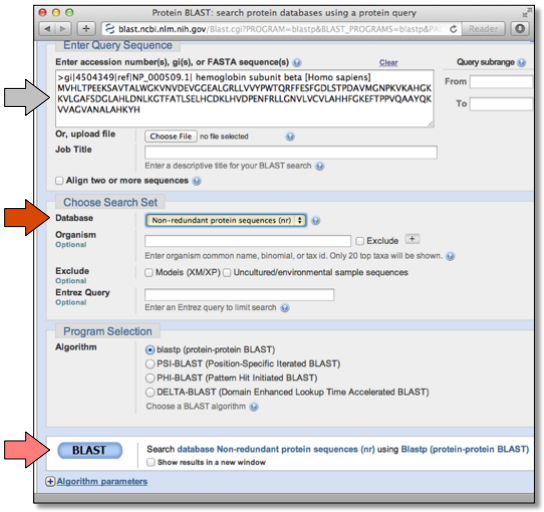


Step 2: Choose the BLAST program



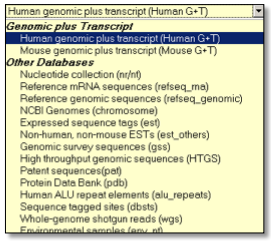
DNA potentially encodes six proteins



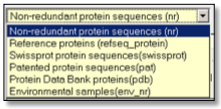


Step 3: Choose the database

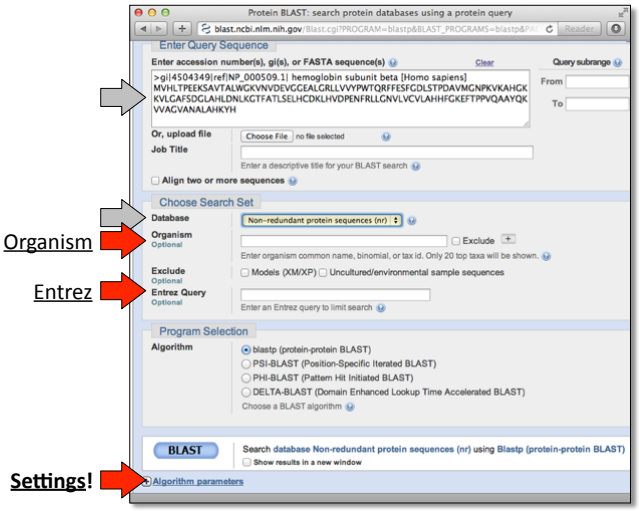
nr = non-redundant (most general database)
dbest = database of expressed sequence tags
dbsts = database of sequence tag sites
gss = genomic survey sequences



nucleotide databases



protein databases

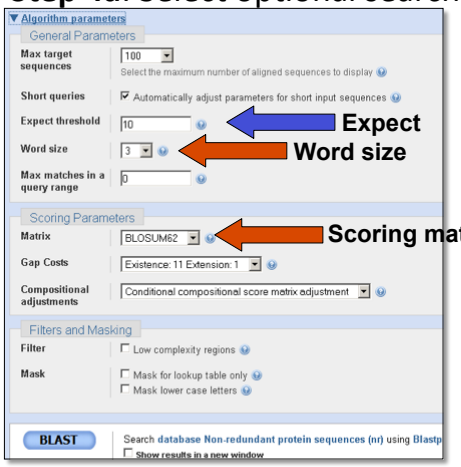


Organism

Entrez

Settings!

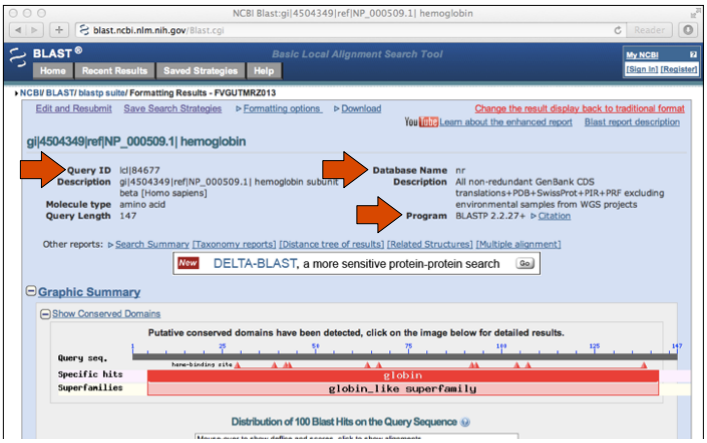
Step 4a: Select optional search parameters



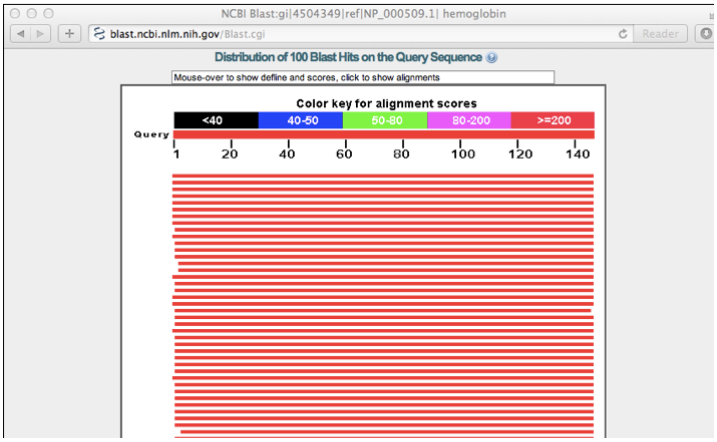
Step 4: Optional parameters

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

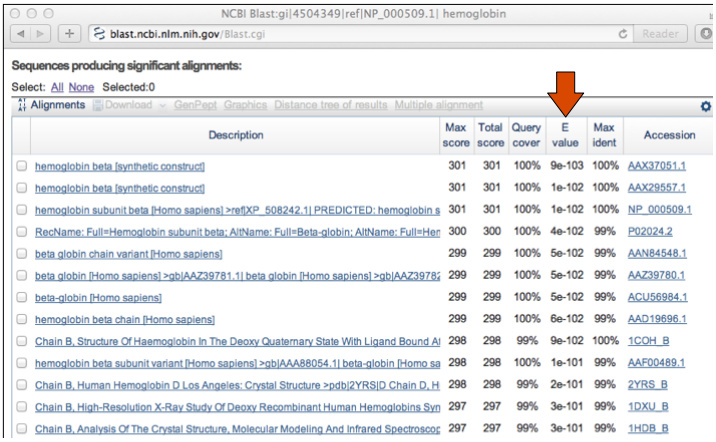
Results page



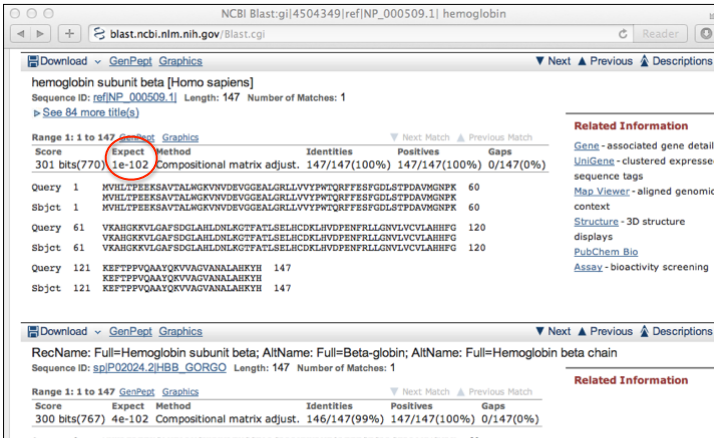
Further down the results page...



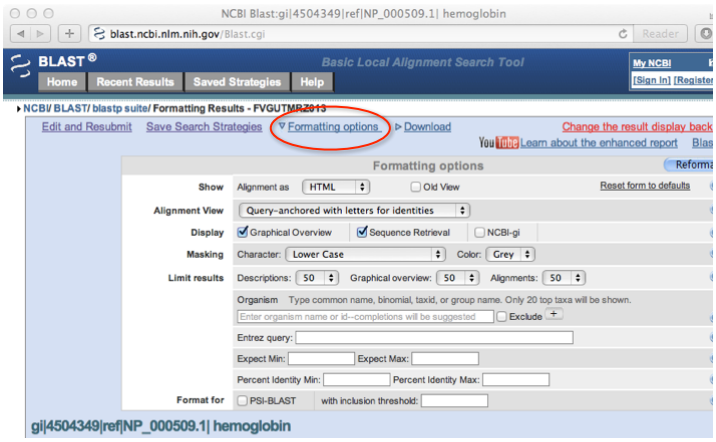
Further down the results page...



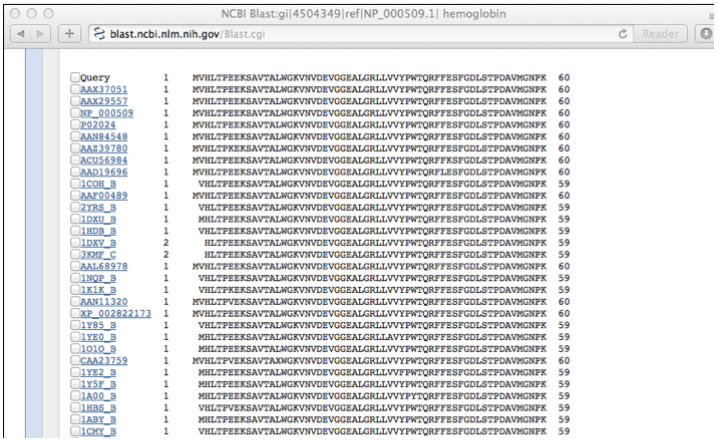
Further down the results page...



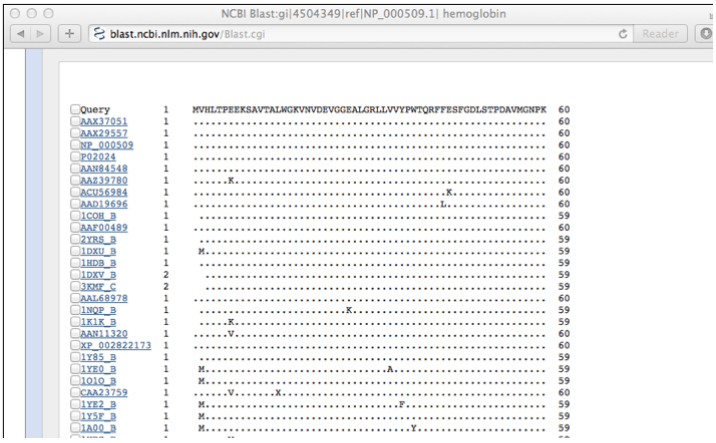
Different output formats are available



E.g. Query anchored alignments



... and alignments with dots for identities



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

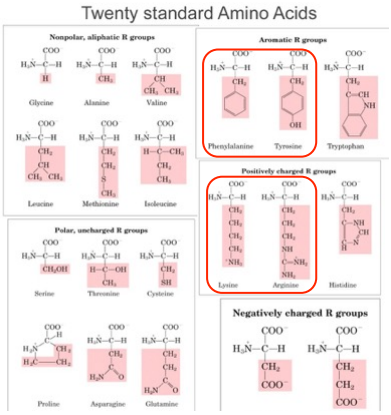
- BLOSUM matrices are based on short, ungapped blocks of conserved amino acid sequences from multiple alignments
  - members of a block that have a most X percent sequence identity to each other are used to generate a BLOSUM**X** matrix
  - For example, using a cutoff of 62% identity will generate the 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	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
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

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

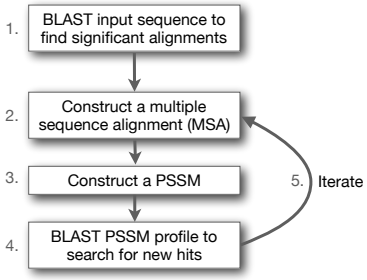
- Use human beta globin as a query against human RefSeq proteins, and blastp does not "find" human myoglobin
  - This is because the two proteins are too distantly related
  - **PSI-BLAST** at NCBI as well as hidden Markov models (HMMs) easily solve this problem
- How can we search using 10,000 base 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



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
```

R,I,K C D,E,T K,R,T N,L,Y,G


```
     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 -3  3  1 -2  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 -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 -2 -1 -1 -1  0 -2 -2 -2 -1 -1 -3 -1  1  0 -3 -2  0
13 W -2 -3 -4 -5 -3 -2 -3 -3 -3 -3 -2 -3 -2  1  4 -3  2  1 -3 -3
14 A  3 -2 -2 -2 -1 -1 -1  0 -2 -2 -2 -1 -1 -3 -1  1 -1 -3 -3 -1
15 A  2 -2 -2 -3 -1 -1 -1  0 -3 -3 -3  0 -2 -3 -1  3  0 -3 -2 -2
16 A  4 -2 -2 -2 -1 -1 -1  0 -2 -2 -2 -1 -1 -3 -1  1  0 -3 -2 -1
...
37 S  2 -2 -2 -3 -2 -1 -2  0 -3 -3 -3  0 -2 -3 -1  4  1 -3 -2 -2
38 G  0 -3 -1 -2 -3 -2 -2  6 -2 -4 -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

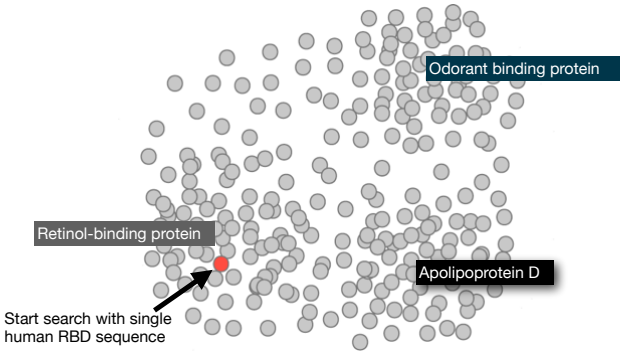
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	M	-1	-2	-2	-3	-2	-1	-2	-3	-2	1	2	-2	6	0	-3	-2	-1	-2	-1	1
2	K	-1	1	0	1	-4	2	4	-2	0	-3	-3	3	-2	-4	-1	0	-1	-3	-2	-3
3	W	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	12	2	-3
4	V	0	-3	-3	-4	-1	-3	-3	-4	-4	3	1	-3	1	-1	-3	-2	0	-3	-1	4
5	W	-3	-3	-4	-5	-3	-2	-3	-3	-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	-1										-3	-1	3	0	-3	-2	-2
16	A	4	-2	-2	-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

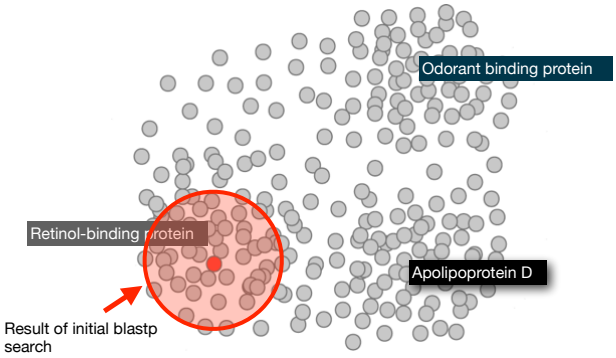
note that a given amino acid (such as alanine) in your query protein can receive different scores for matching alanine—depending on the position in the protein

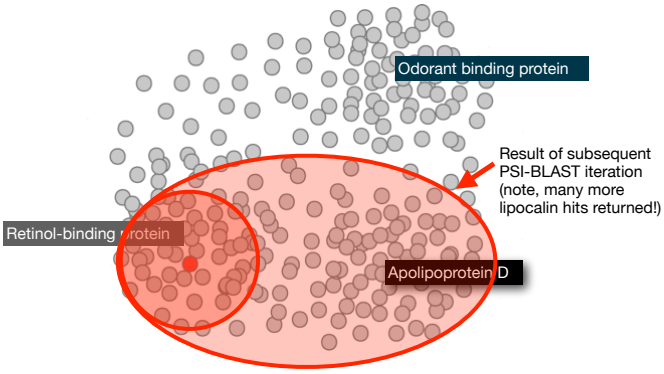
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	M	-1	-2	-2	-3	-2	-1	-2	-3	-2	1	2	-2	6	0	-3	-2	-1	-2	-1	1
2	K	-1	1	0	1	-4	2	4	-2	0	-3	-3	3	-2	-4	-1	0	-1	-3	-2	-3
3	W	-3	-3	-4	-5	-3	-2	-3	-3	-3	-3	-2	-3	-2	1	-4	-3	-3	12	2	-3
4	V	0	-3	-3	-4	-1	-3	-3	-4	-4	3	1	-3	1	-1	-3	-2	0	-3	-1	4
5	W	-3	-3	-4	-5	-3	-2	-3	-3	-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	-1										-3	-1	3	0	-3	-2	-2
16	A	4	-2	-2	-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

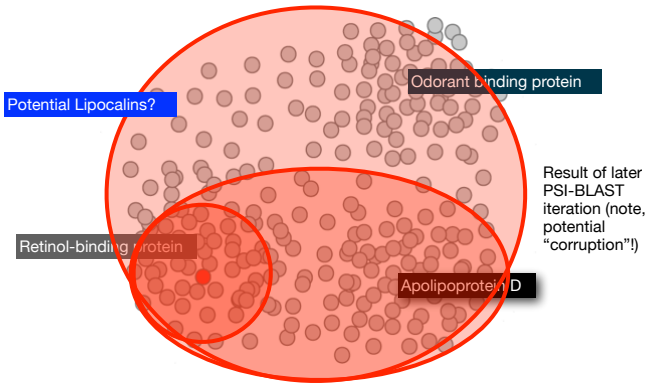
The PSI-BLAST PSSM is essentially a query customized scoring matrix that is more sensitive than PAM or BLOSUM.

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







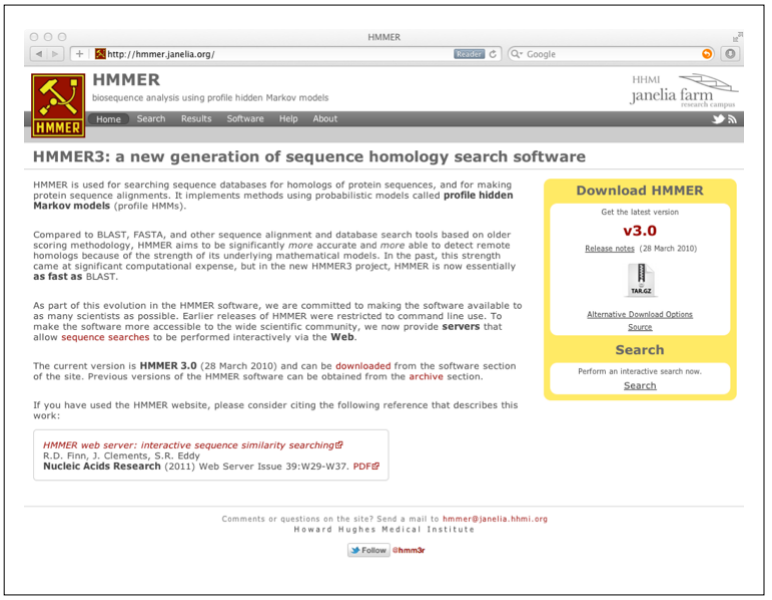


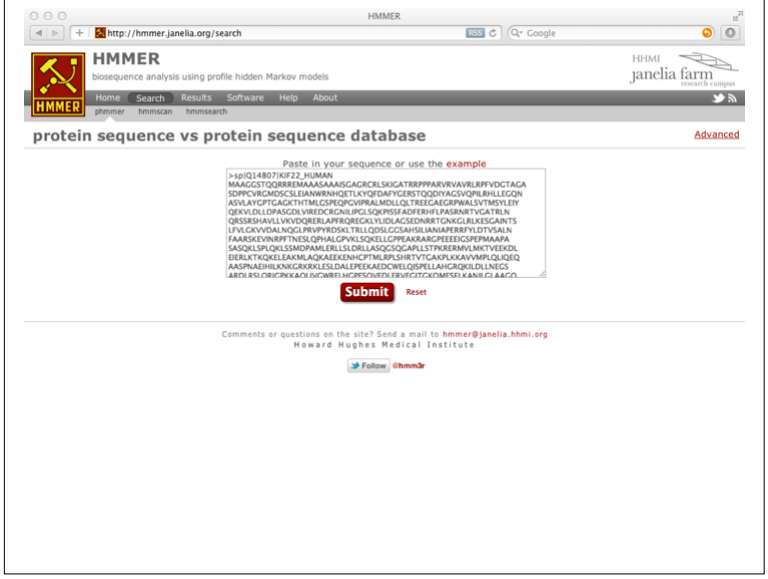
PSI-BLAST returns dramatically more hits

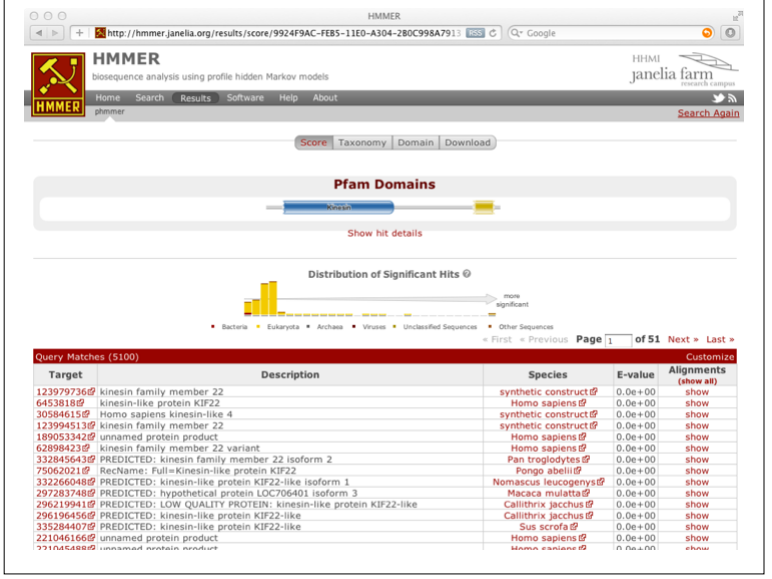
- The search process is continued iteratively, typically about five times, and at each step a new PSSM is built
  - You must decide how many iterations to perform and which sequences to include!
  - You can stop the search process at any point - typically whenever few new results are returned or when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

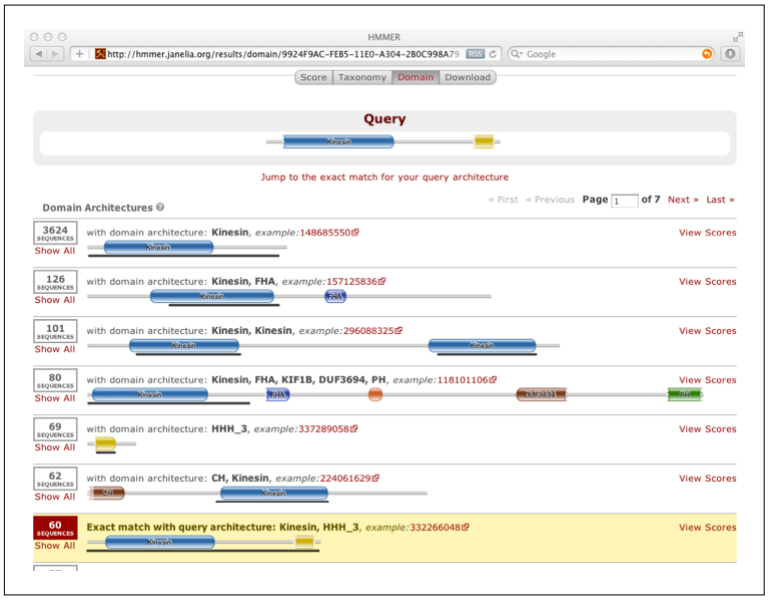
Iteration	Hits with E < 0.005	Hits with E > 0.005
1	34	61
2	314	79
3	416	57
4	432	50
5	432	50

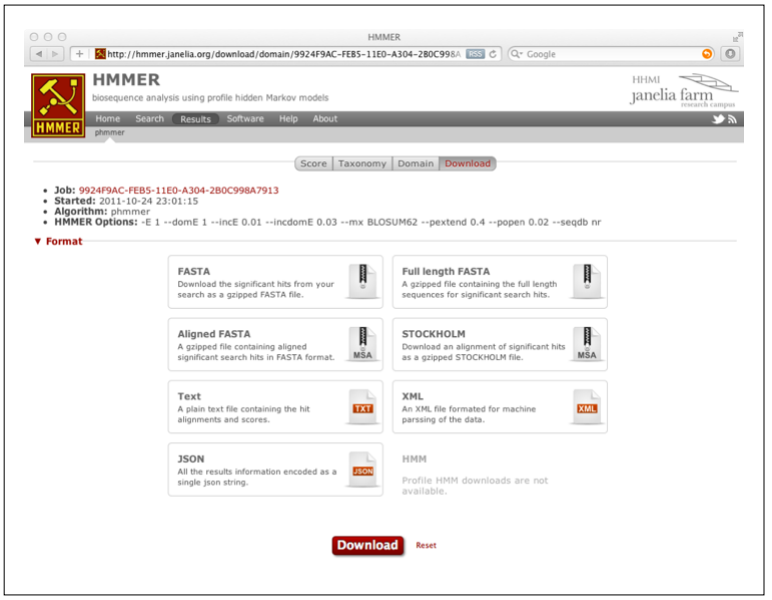
Human retinol-binding protein 4 (RBP4; P02753) was used as a query in a PSI-BLAST search of the RefSeq database.











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