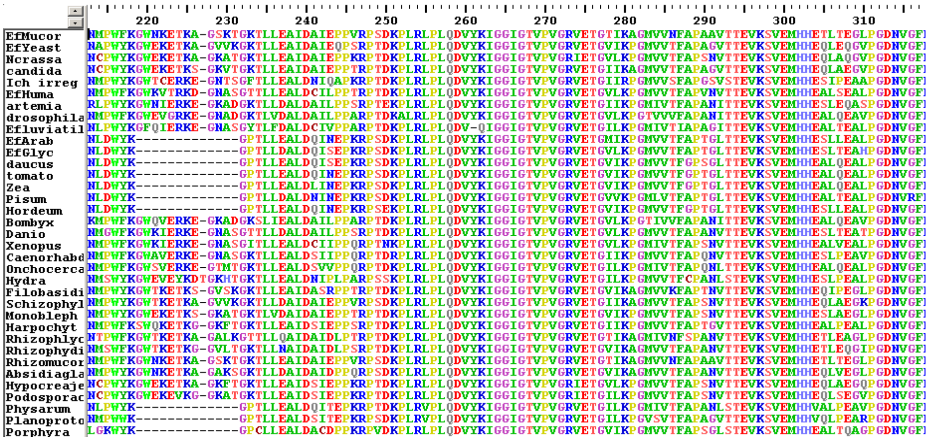
## Multiple Sequence Alignment



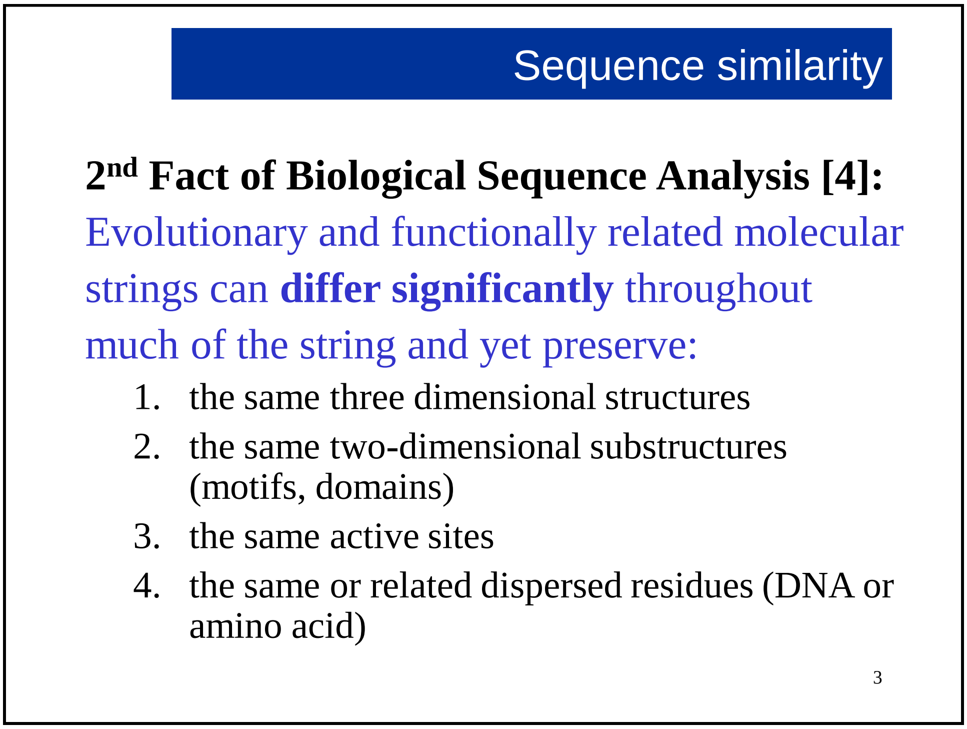
```
           220       230       240       250       260       270       280       290       300       310
EfMucor    NMPWFKGWNKETKA-GSKTGKTLLEAIDAIEPPVRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGTIKAGMVVNFAPAAVTTEVKSVEMHHETLTEGLPGDNVGF
EfYeast    NAPWYKGWEKETKA-GVVKGKTLLEAIDAIEQPSRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFAPAGVTTEVKSVEMHHEQLEQGVPGDNVGF
Ncrassa    NCPWYKGWEKETKA-GKATGKTLLEAIDAIEPPKRPTDKPLRLPLQDVYKIGGIGTVPVGRIETGVLKPGMVVTFAPSNVTTEVKSVEMHHEQLAQGVPGDNVGF
candida    NCPWYKGWEKETKS-GKVTGKTLLEAIDAIEPPTRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGIIKAGMVVTFAPAGVTTEVKSVEMHHEQLAEGVPGDNVGF
Ich irreg  NMPWYKGWTCERKE-GNTSGFTLLEALDNIQAPKRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGIIRPGMVVSFAPGSVSTEVKSVEMHHESIPEAAPGDNVGF
EfHuma     NMPWFKGWKVTRKD-GNASGTTLLEALDCILPPTRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVLKPGMVVTFAPVNVTTEVKSVEMHHEALSEALPGDNVGF
artemia    RLPWYKGWNIERKE-GKADGKTLLDALDAILPPSRPTEKPLRLPLQDVYKIGGIGTVPVGRVETGIIKPGMIVTFAPANITTEVKSVEMHHESLEQASPGDNVGF
drosophila NMPWFKGWEVGRKE-GNADGKTLVDALDAILPPARPTDKALRLPLQDVYKIGGIGTVPVGRVETGVLKPGTVVVFAPANITTEVKSVEMHHEALQEAVPGDNVGF
Efluviatil NLPWYKGFQIERKE-GNASGYTLFDALDCIVPPARPTDKPLRLPLQDV-QIGGIGTVPVGRVETGILKPGMIVTIAPAGITTEVKSVEMHHEALTEALPGDNVGF
EfArab     NLDWYK-------------GPTLLEALDQINEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGMIKPGMVVTFAPTGLTTEVKSVEMHHESLLEALPGDNVGF
EfGlyc     NLDWYK-------------GPTLLDALDQISEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVLKPGMVVTFAPTGLTTEVKSVEMHHESLTEAHPGDNVGF
daucus     NLDWYK-------------GPTLLEALDQISEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFGPSGLTTEVKSVEMHHEALQEALPGDNVGF
tomato     NLDWYK-------------GPTLLEALDQINEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFGPTGLTTEVKSVEMHHEALQEALPGDNVGF
Zea        NLDWYK-------------GPTLLEALDLINEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFGPTGLTTEVKSVEMHHEALQEALPGDNVGF
Pisum      NLDWYK-------------GPTLLDALDNINEPKRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVVKPGMLVTFAPTGLTTEVKSVEMHHEALTEALPGDNVRF
Hordeum    NLDWYK-------------GPTLLEALDQINEPKRPSEKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFGPTGLTTEVKSVEMHHESLLEALPGDNVGF
Bombyx     KMPWFKGWQVERKE-GKADGKSLIEALDAILPPARPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVLKPGTIVVFAPANITTEVKSVEMHHEALQEAVPGDNVGF
Danio      NMGWFKGWKIERKE-GNASGTTLLDALDAILPPSRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVLKPGMVVTFAPANVTTEVKSVEMHHESLTEATPGDNVGF
Xenopus    NMPWFKGWKIERKE-GNASGITLLEALDCIIPPQRPTNKPLRLPLQDVYKIGGIGTVPVGRVETGVLKPGMIVTFAPSNVTTEVKSVEMHHEALVEALPGDNVGF
Caenorhabd NMPWFKGWAVERKE-GNASGKTLLEALDSIIPPQRPTDRPLRLPLQDVYKIGGIGTVPVGRVETGIIKPGMVVTFAPQNVTTEVKSVEMHHESLPEAVPGDNVGF
Onchocerca NMPWFKGWSVERKE-GTMTGKTLLEALDSVVPPQRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGILKPGMIVTFAPQNLTTEVKSVEMHHEALQEALPGDNVGF
Hydra      NMSWYKGWEVEYKDTGKHTGKTLLEALDNIPLPARPSSKPLRLPLQDVYKIGGIGTVPVGRVETGILKPGMVVTFCPANLSTEVKSVEMHHESLPEALPGDNVGF
Filobasidi NMPWYKGWTKETKS-GVSKGKTLLEAIDASRPPTRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKAGMVVKFAPTNVTTEVKSVEMHHEQIPEGLPGDNVGF
Schizophyl NMPWYKGWTKETKA-GVVKGKTLLDAIDAIEPPVRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGIIKAGMVVTFAPSNVTTEVKSVEMHHEQLAEGKPGDNVGF
Monobleph  NMPWYKGWEKETKS-GKATGKTLVDAIDAIEPPTRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFAPSNVTTEVKSVEMHHESLAEGLPGDNVGF
Harpochyt  NMPWFKSWQKETKG-GKFTGKTLLEAIDSIEPPSRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGIIKPGMVVTFAPTGVTTEVKSVEMHHEALPEALPGDNVGF
Rhizophlyc NTPWFKGWTKETKA-GALKGTTLLQAIDAIDLPTRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGTIKAGMIVNFSPANVTTEVKSVEMHHETLEAGLPGDNVGF
Rhizophydi NMSWFKGWTKETKG-GVLTGKTLLNAIDAIDLPSRPTDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKAGMVVTFAPANVTTEVKSVEMHHETLEQGIPGDNVGF
Rhizomucor NMPWFKGWNKETKA-GSKTGKTLLEAIDAIEPPVRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGTIKAGMVVNFAPAAVTTEVKSVEMHHETLTEGLPGDNVGF
Absidiagla NMPWYKGWNKETKA-GAKSGKTLLDAIDAIDPPQRPSDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKAGMVVTFAPANVTTEVKSVEMHHEQLVEGLPGDNVGF
Hypocreaje NCPWYKGWEKETKA-GKFTGKTLLEAIDSIEPPKRPTDKPLRLPLQDVYKIGGIGTVPVGRIETGVLKPGMVVTFAPSNVTTEVKSVEMHHEQLAEGQPGDNVGF
Podosporac NCPWYKGWEKEVKG-GKATGKTLLEAIDSIEPPKRPTDKPLRLPLQDVYKIGGIGTVPVGRIETGILKPGMVVTFAPSNVTTEVKSVEMHHEQLSEGVPGDNVGF
Physarum   NLPWYK-------------GPTLLEALDQITEPKRPTDKPLRVPLQDVYKIGGIGTVPVGRVETGILKPGMIVTFAPANLSTEVKSVEMHHVALPEAVPGDNVGF
Planoproto NMPWWK-------------GPTLLEALDSITEPKRPSDKPLRVPLQDVYKIGGIGTVPVGRVETGILKPGVSVTFAPAGVTTEVKSVEMHHVQLPEARPGDNVGF
Porphyra   LGKWYK-------------GPCLLEALDACDPPKRPVDKPLRLPLQDVYKIGGIGTVPVGRVETGVIKPGMVVTFAPSGLSTEVKSVEMHHEALTQAGPGDNVGF
```

## Multiple Sequence Alignment

Shows multiple similarities:

- Common structure of protein product
- Common function
- Common evolutionary process

- Protein Structure Prediction
- Protein Family Identification
- Protein Characterization: signatures of protein families
- Phylogeny estimation

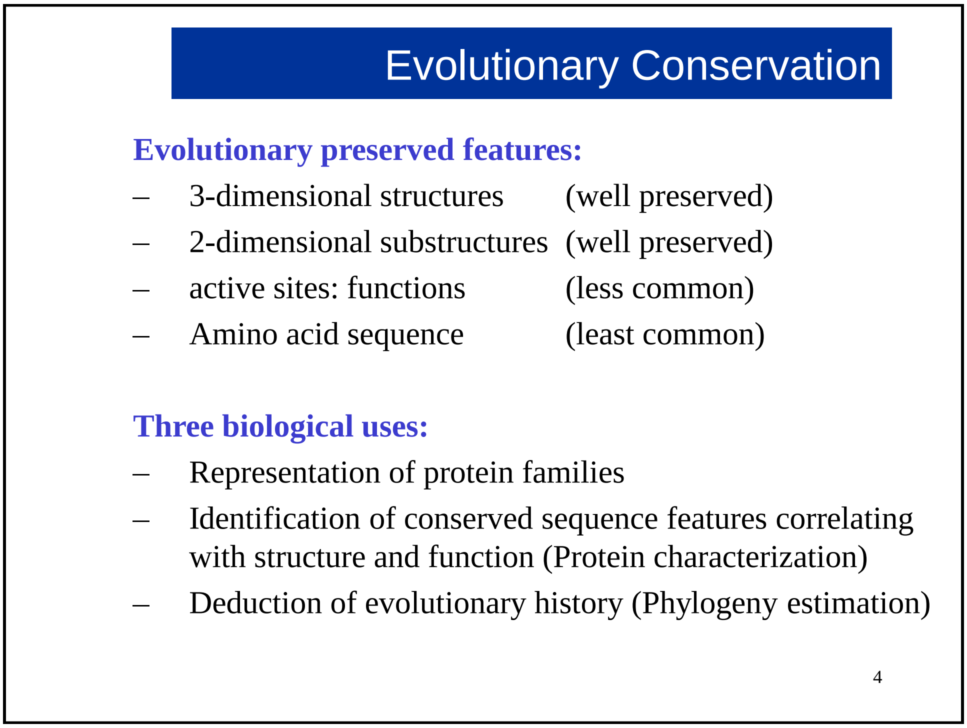


## Sequence similarity

**2nd Fact of Biological Sequence Analysis [4]:**

Evolutionary and functionally related molecular strings can **differ significantly** throughout much of the string and yet preserve:

1. the same three dimensional structures
2. the same two-dimensional substructures (motifs, domains)
3. the same active sites
4. the same or related dispersed residues (DNA or amino acid)



## Evolutionary Conservation

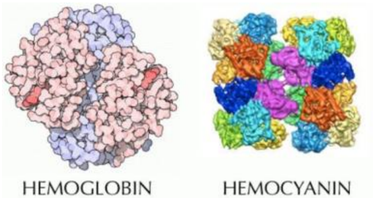
**Evolutionary preserved features:**

| | |
|---|---|
| – 3-dimensional structures | (well preserved) |
| – 2-dimensional substructures | (well preserved) |
| – active sites: functions | (less common) |
| – Amino acid sequence | (least common) |

**Three biological uses:**

- Representation of protein families
- Identification of conserved sequence features correlating with structure and function (Protein characterization)
- Deduction of evolutionary history (Phylogeny estimation)

## An Amazing Example: Hemoglobin

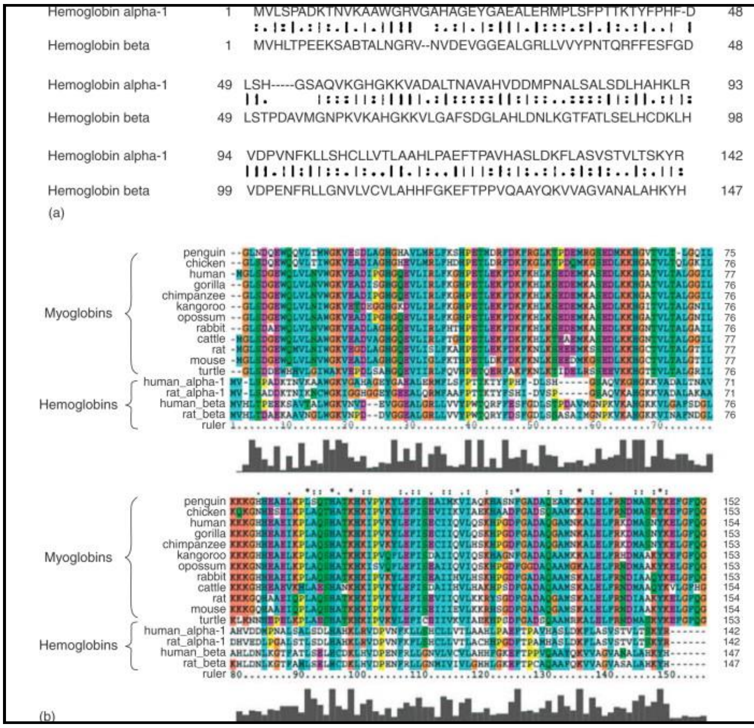


**Hemoglobin:**

- An almost universal protein found in birds, mammals, etc.
- 4 chains of ~140 amino acids
- Functions the same in all birds, mammals, etc.: binds and transports oxygen
- Insects and mammals diverged ~600 million years ago
  => On average 100 amino acids mutations per chain
- Pair wise alignment:

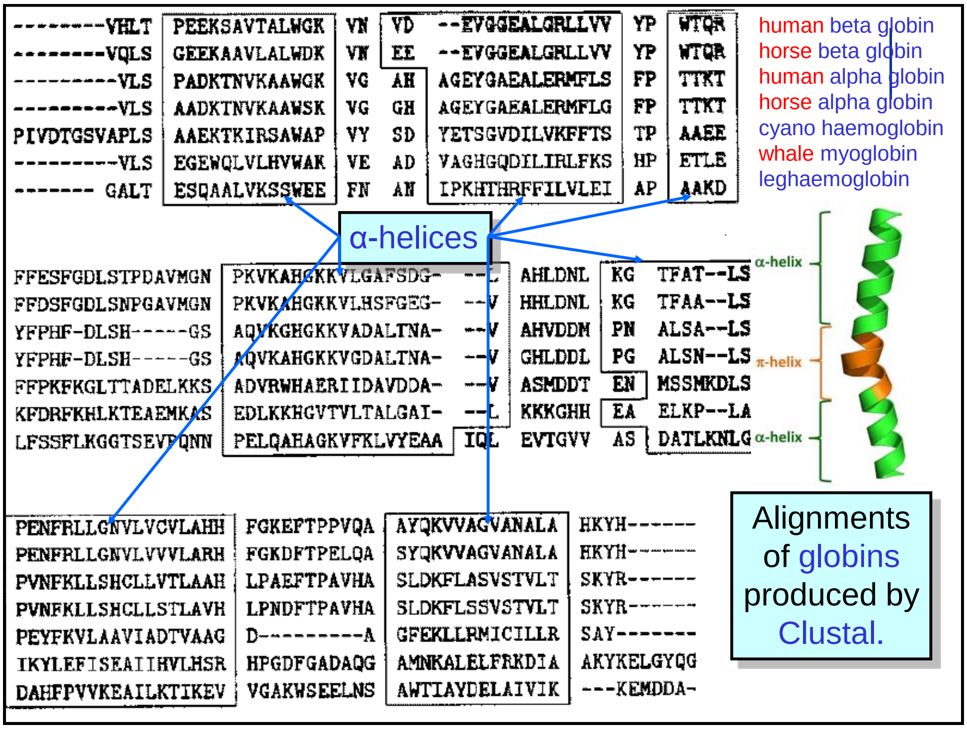
| | |
|---|---|
| – human - chimpanzee | equal |
| – Mammal - mammal | suggests functional similarity |
| – Insect-mammal | very little similarity! |

- Secondary and 3-dimensional structure well preserved



```
Hemoglobin alpha-1    1   MVLSPADKTNVKAAWGRVGAHAGEYGAEALERMPLSFPTTKTYFPHF-D   48
Hemoglobin beta       1   MVHLTPEEKSABTALNGRV--NVDEVGGEALGRLLVVYPNTQRFFESFGD  48

Hemoglobin alpha-1   49  LSH-----GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR   93
Hemoglobin beta      49  LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH   98

Hemoglobin alpha-1   94   VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR  142
Hemoglobin beta      99   VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH  147
(a)
```

From [14].

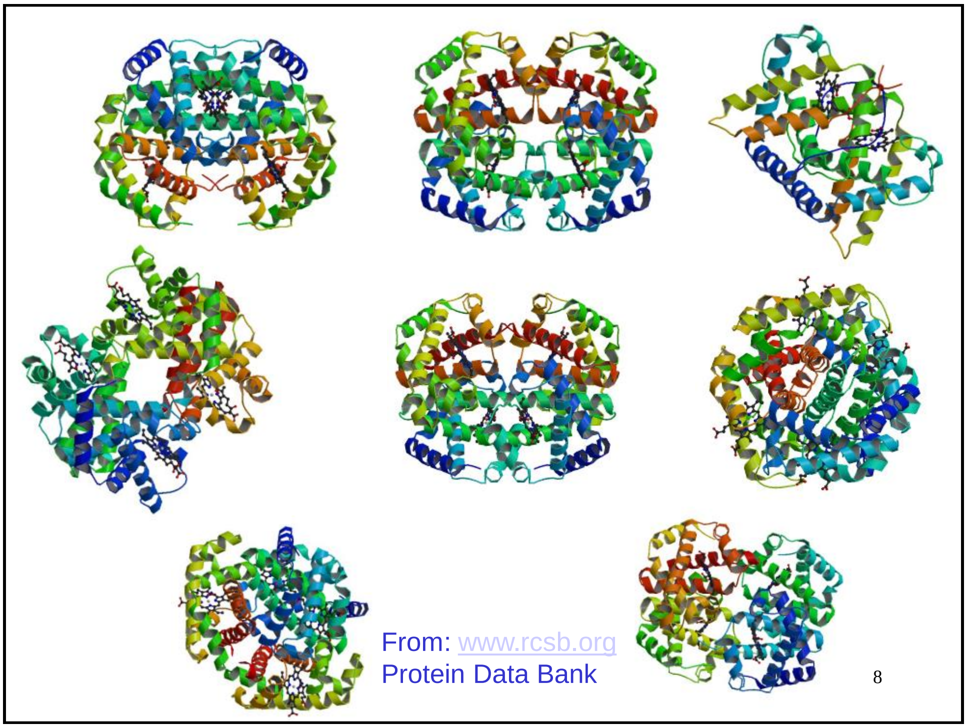


```
---------VHLT  PEEKSAVTALWGK  VN  VD  --EVGGEALGRLLVV  YP  WTQR
---------VQLS  GEEKAAVLALWDK  VN  EE  --EVGGEALGRLLVV  YP  WTQR
----------VLS  PADKTNVKAAWGK  VG  AH  AGEYGAEALERMFLS  FP  TTKT
----------VLS  AADKTNVKAAWSK  VG  GH  AGEYGAEALERMFLG  FP  TTKT
PIVDTGSVAPLS   AAEKTKIRSAWAP  VY  SD  YETSGVDILVKFFTS  TP  AAEE
----------VLS  EGEWQLVLHVWAK  VE  AD  VAGHGQDILIRLFKS  HP  ETLE
-------- GALT  ESQAALVKSSWEE  FN  AN  IPKHTHRFFILVLEI  AP  AAKD

FFESFGDLSTPDAVMGN  PKVKAHGKKVLGAFSDG-  --L  AHLDNL  KG  TFAT--LS
FFDSFGDLSNPGAVMGN  PKVKAHGKKVLHSFGEG-  --V  HHLDNL  KG  TFAA--LS
YFPHF-DLSH-----GS  AQVKGHGKKVADALTNA-  --V  AHVDDM  PN  ALSA--LS
YFPHF-DLSH-----GS  AQVKAHGKKVGDALTNA-  --V  GHLDDL  PG  ALSN--LS
FFPKFKGLTTADELKKS  ADVRWHAERIIDAVDDA-  --V  ASMDDT  EN  MSSMKDLS
KFDRFKHLKTEAEMKAS  EDLKKHGVTVLTALGAI-  --L  KKKGHH  EA  ELKP--LA
LFSSFLKGGTSEVPQNN  PELQAHAGKVFKLVYEAA  IQL  EVTGVV  AS  DATLKNLG

PENFRLLGNVLVCVLAHH  FGKEFTPPVQA  AYQKVVAGVANALA  HKYH------
PENFRLLGNVLVVVLARH  FGKDFTPELQA  SYQKVVAGVANALA  HKYH------
PVNFKLLSHCLLVTLAAH  LPAEFTPAVHA  SLDKFLASVSTVLT  SKYR------
PVNFKLLSHCLLSTLAVH  LPNDFTPAVHA  SLDKFLSSVSTVLT  SKYR------
PEYFKVLAAVIADTVAAG  D---------A  GFEKLLRMICILLR  SAY-------
IKYLEFISEAIIHVLHSR  HPGDFGADAQG  AMNKALELFRKDIA  AKYKELGYQG
DAHFPVVKEAILKTIKEV  VGAKWSEELNS  AWTIAYDELAIVIK  ---KEMDDA-
```

human beta globin
horse beta globin
human alpha globin
horse alpha globin
cyano haemoglobin
whale myoglobin
leghaemoglobin

α-helices

Alignments of globins produced by Clustal.



From: www.rcsb.org
Protein Data Bank

## Multiple Alignment Examples

**Related sequences** can have so few conserved or so dispersed matching amino acids: statistically indistinguishable from the best alignment of 2 random strings.

For example this is true for:

- Hemoglobin; immunoglobulin (antibody proteins)
- E-Cadherin (adhesion molecule)

Compare to:

1 DNA nucleotide change:

- Sickle cell anemia

## (Global) Multiple Alignment

Definition

A *multiple alignment* of a set of strings $\{S_1,S_2,\ldots,S_k\}$ is a series of strings $S`_1,S`_2, \ldots ,S`_k$ such that

1. $|S`_1| = |S`_2| = \ldots = |S`_k|$ (all $S`_i$ have the same length)
2. For every $i$: $S`_i$ is an extension of $S_i$ obtained by insertion of spaces.

A multiple alignment of strings ACBCBD, CADDB, and ACABCD:

```
AC..BCDB
.CADB.D.
ACA.BCD.
```

Note: similarly, local *multiple alignment* (for substrings) can be defined.

## Multiple Alignment

Definition **Induced Pair-Wise Alignment:**

Given a multiple alignment M, the induced pair-wise alignment of two string $S_i$ and $S_j$ is obtained from M by removing all rows except rows i an j.

Note: two opposing spaces can be removed

Definition **Score of Induced Pair-Wise Alignment**:

The score of an induced pair-wise alignment is determined using any chosen *scoring scheme* for 2-string alignment in the standard manner.

## Multiple Alignment: Sum-of-Pairs

Definition: **Sum of Pairs Score**

The sum-of-pairs (SP) score of a multiple alignment M is the sum of the scores of pair-wise global alignments *induced* by M.

Problem: **The SP Alignment Problem**

Compute a global multiple alignment M with minimum sum-of-pairs score.

**Note:** Intuitively this is a reasonable score,
but: **no theoretical justification**!

## An Exact Solution of the SP Alignment Problem

### A Dynamic Programming Approach

For the calculation of the best **multiple alignment of *r* sequences** an *r-dimensional hypercube* ***D*** can be used, where:

- Each dimension is formed by one of the *r* sequences.
- The nodes $(j_1, j_2, \ldots, j_r)$ of the hypercube hold the best score $\boldsymbol{D(j_1, j_2, \ldots, j_r)}$ for aligning the prefixes of the sequences $S_1, S_2, \ldots, S_r$ of lengths $j_1, j_2, \ldots, j_r$, respectively.

## Multiple Alignment: Dynamic Programming

The best score *D* can be calculated using dynamic programming on the following recursion:

$$D(0,0,\ldots,0) = 0$$

$$D(j_1, j_2, \ldots, j_r) = \min_{\varepsilon \in \{0,1\}^r, \varepsilon \neq 0} \{D(j_1 - \varepsilon_1, j_2 - \varepsilon_2, \ldots, j_r - \varepsilon_r) + \rho(\varepsilon_1 x_{j_1}, \varepsilon_2 x_{j_2}, \ldots, \varepsilon_r x_{j_r})\}$$

and $\rho$ the cost function (for example Sum of Pairs (SP))

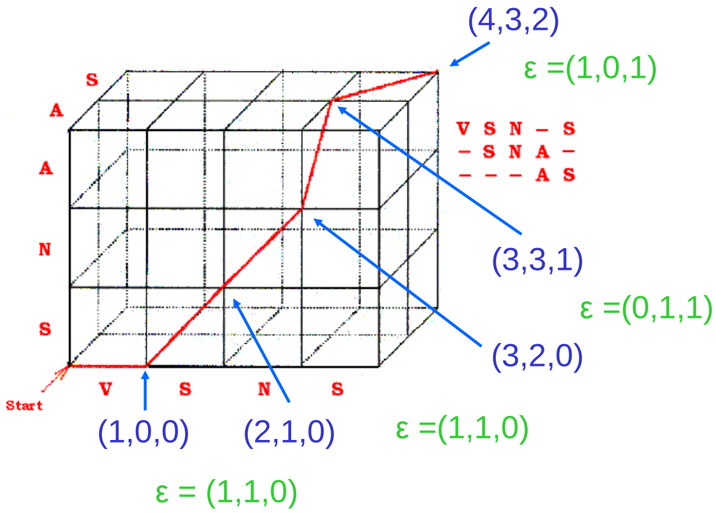
$$\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \in \{0,1\}^r$$

$$\varepsilon \neq (1,1,\ldots,1)$$

Note:

- There is no generally accepted score for a multiple alignment. We will discuss some candidates.
- Biological relevance of the multiple alignment is of major importance.

## Multiple Alignment: Dynamic Programming



Space complexity: $O(n^r)$.(O(r-dimensional cost function ρ calculation))
Time complexity: $O(2^r n^r)$ (consider $2^r$-1 previously calculated values)

## Exact Multiple Alignment Complexity

The **Exact Multiple Alignment problem** solved by Dynamic Programming has:

Space complexity $O(n^r)$ . (O( *calculation of ρ* ))
Time complexity $O(2^r n^r)$

This exact solution using dynamic programming is only useful for a small number of strings, i.e., a small r.

In general:

The exact multiple alignment problem using *sum-of-pairs* or *evolutionary-tree* scoring is NP-complete [9].

## Scoring Metrics

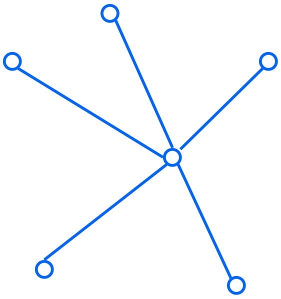
**Possible $\rho$ functions:**

- Sum of Pairs
  - Defined as the sum of pairwise distances between all pairs of sequences: $$\sum_{i,j\in\{1,\ldots,r\},i<j} D(S_i,S_j)$$
- Distance from Consensus
  - Let C be the consensus sequence, then the total distance is defined as: $$\sum_{i\in\{1,\ldots,r\}} D(S_i,C)$$
- Evolutionary Tree Alignment
  - Define *$D(S_v,S_w)$* as the number of changes between $S_v$ and $S_w$, then the weight of an evolutionary tree is defined as: $$\sum_{(v,w)\in T} D(S_v,S_w)$$
  - Then the weight of the lightest evolutionary tree that can be constructed from the sequences is taken.

## Approximation: The Center Star Method for (SP) Alignment.

- An **approximation** algorithm for the optimal alignment under the *Sum of Pairs metric* [4, pp348-350] achieving approximation ratio of *2*.
- Sum of Pairs metric
  - Defined as the sum of induced pair-wise distances between all pairs of sequences: $$\sum_{i,j\in\{1,\ldots,r\},i<j} D(S_i,S_j)$$
  - Where *$D(S,T)$* the score of induced alignment of *$S$* with *$T$*, using a scoring function *$d(x,y)$* such that:
    - *$d(-,-)=0$*
    - *$d(x,y) = d(y,x)$*
    - *$d(x,y) \leq d(x,z) + d(z,y)$* **(triangle inequality)**

## star alignment



- We put a string in center (which one?)
- Compute pair-wise alignments
- Extend pairs into multiple alignment
- Where “once a gap, always a gap”

Complexity: O($k^2n^2$)  k sequences, n length

Theory proves: within factor 2 of optimal alignment

## Star Alignment

Problem: The Sum of Pairs alignment problem

Input: A set of sequences $\{S_1, S_2, \ldots, S_k\}$

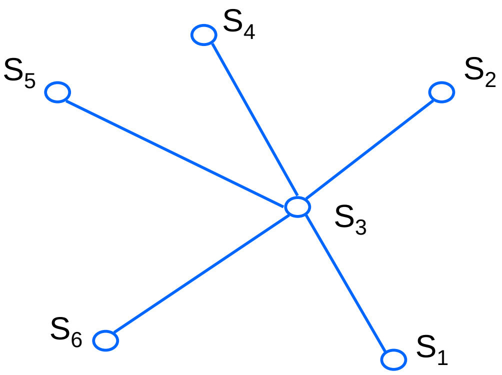
Question: Compute a global multiple alignment $M$ with minimum sum-of-pairs score.

Definition: Given a set of $k$ strings $S$, define a center string $S_c$ (element of $S$) as a string that minimizes: $\sum_{S_j \in S} D(S_c, S_j)$

Definition: Define the **center star** to be a star tree of k nodes, with the center node $S_c$ and with each of the remaining $k$-$1$ nodes labeled by a distinct string from $S \backslash \{S_c\}$.

Definition: Define the multiple alignment $M_c$ of the set of strings $S$ to be the multiple alignment consistent with the center star.

## The Center Star Method for (SP) Alignment.

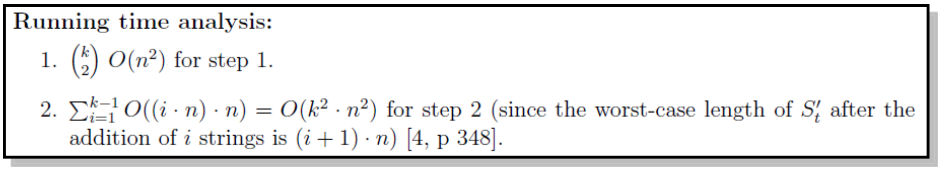


A generic center start for 6 strings with center string $S_C = S_3$. [4, p349].

## Star Alignment

### The Center Star Algorithm

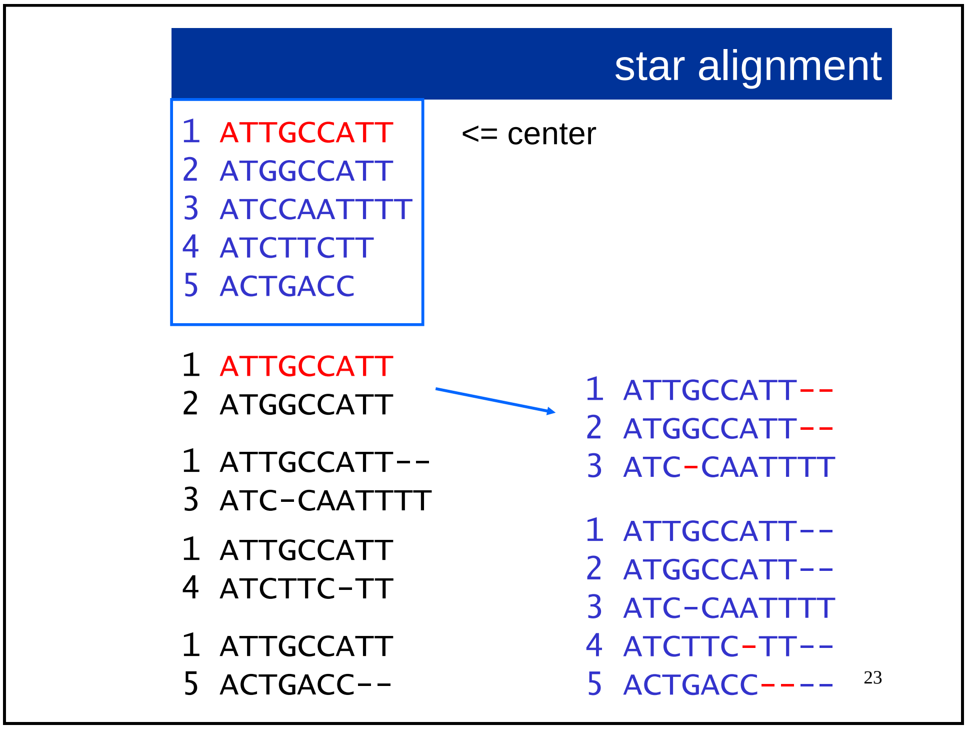
1. Find $S_t$ from $S$ minimizing $\sum_{i \neq t} D(S_i, S_t)$ and let $M=\{S_t\}$.
2. Add the sequences in $S \backslash \{S_t\}$ to $M$ one by one such that
   - the alignment of every newly added sequence with $S_t$ is optimal.
   - Add spaces when needed, to all pre-aligned sequences.



**Running time analysis:**

1. $\binom{k}{2}$ $O(n^2)$ for step 1.

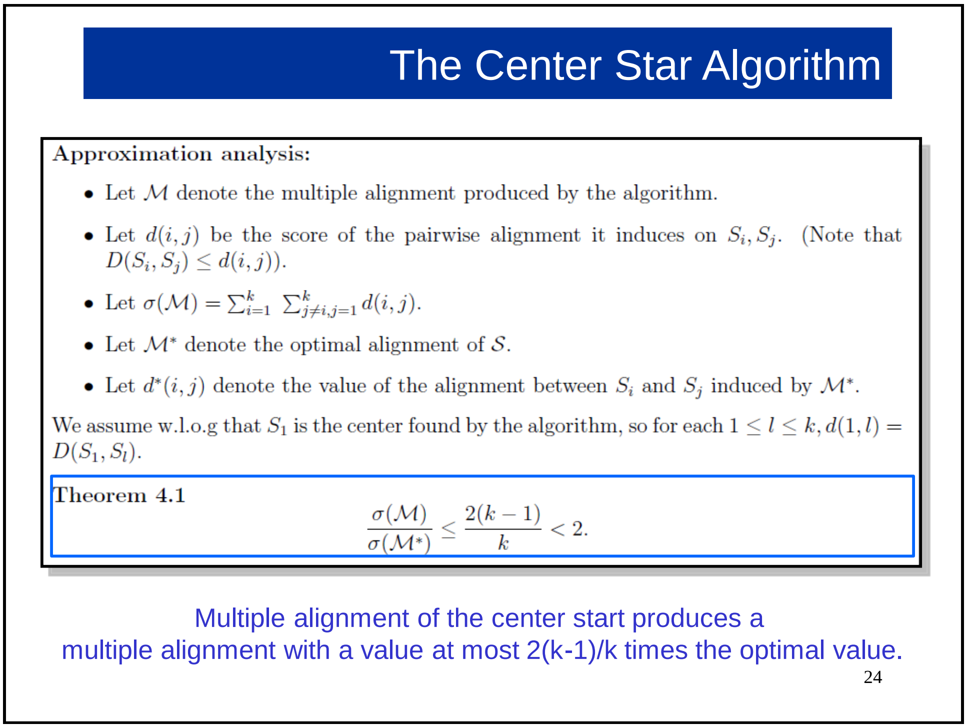
2. $\sum_{i=1}^{k-1} O((i \cdot n) \cdot n) = O(k^2 \cdot n^2)$ for step 2 (since the worst-case length of $S_t'$ after the addition of $i$ strings is $(i+1) \cdot n$) [4, p 348].



## star alignment

```
1  ATTGCCATT       <= center
2  ATGGCCATT
3  ATCCAATTTT
4  ATCTTCTT
5  ACTGACC
```

```
1  ATTGCCATT
2  ATGGCCATT          ->     1  ATTGCCATT--
                             2  ATGGCCATT--
1  ATTGCCATT--               3  ATC-CAATTTT
3  ATC-CAATTTT
                             1  ATTGCCATT--
1  ATTGCCATT                 2  ATGGCCATT--
4  ATCTTC-TT                 3  ATC-CAATTTT
                             4  ATCTTC-TT--
1  ATTGCCATT                 5  ACTGACC----
5  ACTGACC--
```



## The Center Star Algorithm

**Approximation analysis:**

- Let $\mathcal{M}$ denote the multiple alignment produced by the algorithm.
- Let $d(i, j)$ be the score of the pairwise alignment it induces on $S_i, S_j$. (Note that $D(S_i, S_j) \leq d(i, j)$).
- Let $\sigma(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j \neq i, j=1}^{k} d(i, j)$.
- Let $\mathcal{M}^*$ denote the optimal alignment of $\mathcal{S}$.
- Let $d^*(i, j)$ denote the value of the alignment between $S_i$ and $S_j$ induced by $\mathcal{M}^*$.

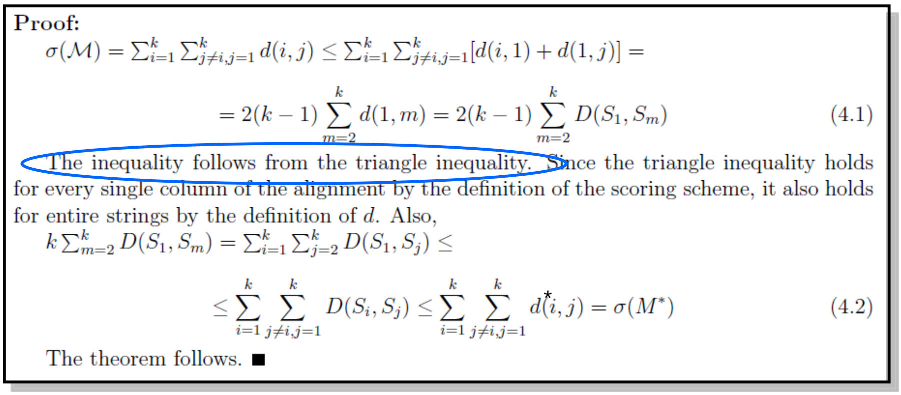
We assume w.l.o.g that $S_1$ is the center found by the algorithm, so for each $1 \leq l \leq k, d(1, l) = D(S_1, S_l)$.

**Theorem 4.1**

$$\frac{\sigma(\mathcal{M})}{\sigma(\mathcal{M}^*)} \leq \frac{2(k-1)}{k} < 2.$$

Multiple alignment of the center start produces a
multiple alignment with a value at most 2(k-1)/k times the optimal value.

# The Center Star Algorithm



**Proof:**

$\sigma(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j \neq i, j=1}^{k} d(i,j) \leq \sum_{i=1}^{k} \sum_{j \neq i, j=1}^{k} [d(i,1) + d(1,j)] =$

$$= 2(k-1) \sum_{m=2}^{k} d(1,m) = 2(k-1) \sum_{m=2}^{k} D(S_1, S_m) \qquad (4.1)$$

The inequality follows from the triangle inequality. Since the triangle inequality holds for every single column of the alignment by the definition of the scoring scheme, it also holds for entire strings by the definition of $d$. Also,

$k \sum_{m=2}^{k} D(S_1, S_m) = \sum_{i=1}^{k} \sum_{j=2}^{k} D(S_1, S_j) \leq$

$$\leq \sum_{i=1}^{k} \sum_{j \neq i, j=1}^{k} D(S_i, S_j) \leq \sum_{i=1}^{k} \sum_{j \neq i, j=1}^{k} d^{*}(i,j) = \sigma(M^*) \qquad (4.2)$$

The theorem follows. ■

Triangle inequality: $D(S_i,S_j) \leq D(S_i,S_1) + D(S_1,S_j)$

# Many Different Alignment Methods

- Aligning a String to a Profile (later HMMs)
- Iterative Pair-wise Alignment
- Progressive Multiple Alignment
  - Feng-Doolittle (1987)[2]
  - CLUSTALW, CLUSTALX
  - State of the Art Parallel MSA (2018) [15]
    - Coffee, MAFT, MSAProbs, M2Align
- PAGAN Phylogeny Aware MSA (2015) [13]
- Etc.