# Chapter 3 Multiple Sequence Alignments

	* . : . * :::.	
Q5E940 BOVIN	MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENNPALE	76
RLA0 HUMAN	MPREDRATWKSNYFLKIIQLLDDYPKCFIYGADNYGSKQMQQIRMSLRGK-AVYLMGKNTMMRKAIRGHLENNPALE	76
RLA0 MOUSE	MPREDRATWKSNYFLKIIQLLDDYPKCFIYGADNYGSKQMQQIRMSLRGK-AYYLMGKNTMMRKAIRGHLENNPALE	76
RLAO_RAT	MPREDRATWKSNYFLKIIQLLDDYPKCFIYGADNYGSKQMQQIRMSLRGK-AYYLMGKNTMMRKAIRGHLENNPALE	76
RLAO CHICK	MPREDRATWKSNYFMKIIQLLDDYPKCFVVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENNPALE	76
RLA0 RANSY	MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENNSALE	76
Q7ZUG3_BRARE	MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQTIRLSLRGK-AVVLMGKNTMMRKAIRGHLENNPALE	76
RLA0 ICTPU	MPREDRATWKSNYFLKIIQLLNDYPKCFIVGADNVGSKQMQTIRLSLRGK-AIVLMGKNTMMRKAIRGHLENNPALE	76
RLA0 DROME	MWRENKAAW <mark>K</mark> AQYFIKWWELFDEF <mark>P</mark> KCFIW <mark>GADNWGS</mark> K <mark>Q</mark> MQNIRTSIRGL-AWWIM <mark>GKNT</mark> MMRKAIRGHIENNPQLE	76
RLA0_DICDI	MSGAG-SKRKKLFIEKATKLFTTYDKMIVAEADFVGSSQLQKIRKSIRGI-GAVLMGKKTMIRKVIRDLADSKPELD	75
Q54LP0_DICDI	MS <mark>G</mark> AG-SKRKNYFIEKATKLFTT YDKMIYAEADFYGSSQLQKIRKSIRGI-GAYLMGKK <mark>TMIRKYIRDL</mark> ADSKPELD	75
RLA0_PLAF8	MAKLSKQQK <mark>K</mark> QMYIEKLSSLIQQ <mark>Y</mark> SKILIYHYDNYG <mark>S</mark> N <mark>Q</mark> MASYRKSL <mark>RG</mark> K-ATILM <mark>GKNT</mark> RIRTALKKNLQAYPQIE	76
RLA0_SULAC	MI <mark>G</mark> LAVTTTKK IAKW <mark>K</mark> VDE VAELT <mark>E</mark> KLKTHKTIIIAN IEGFPADKLHE I <mark>R</mark> KKL <mark>RG</mark> K - ADIK VTKNNLFN IALKNAG YDTK	79
RLA0_SULTO	MRIMAVITQERKIAKW <mark>K</mark> IEEVKELE <mark>Q</mark> KLRE <mark>Y</mark> HTIIIANI <mark>EGFP</mark> ADKLHDI <mark>R</mark> KKM <mark>RG</mark> M-AEIKV <mark>TKNT</mark> LF <mark>G</mark> IAAKNAGLDVS	80
RLA0_SULSO	MKRLALALKQRKVASW <mark>K</mark> LEEVKELT <mark>E</mark> LIKNSNTILI <mark>G</mark> NL <mark>EGFP</mark> ADKLHEI <mark>R</mark> KKL <mark>RG</mark> K-A <mark>TIKVTKNT</mark> LFKIAAKNAGIDIE	80
RLA0_AERPE	MSVVSLV <mark>G</mark> QMYKREK <mark>PIPEWK</mark> TLMLRELE <mark>E</mark> LFSKHRVVLFADLT <mark>GTPT</mark> FVV <mark>Q</mark> RV <mark>R</mark> KKLWKK- <mark>YP</mark> MMVA <mark>KKRIILRAMK</mark> AAGLELDDN	86
RLA0_PYRAE	-MMLAI <mark>G</mark> KRRYØRTRQ <mark>YP</mark> ARKØKIVSEAT <mark>E</mark> LLQK <mark>YP</mark> YØFLFDLH <mark>G</mark> LS <mark>S</mark> RILHE <mark>YR</mark> YRL <mark>R</mark> RY- <mark>GVIKIIKPT</mark> LFKIAFTKV <mark>YGG</mark> IPAE	85
_	MAEERHHTEH I <mark>P</mark> QWKKDE IEN IKEL IQSHKVFGMVG IEG ILATKMQK IRRDLKDV-AVLKV <mark>SRNTLT</mark> ERALNQLGET IP	78
	MAEERHHTEH I <mark>P</mark> QWKKDE IEN IKEL IQSHKVFGMVR IEG ILATK IQK IRRD LKDV-AVLKV <mark>SRNTLT</mark> ERALNQLGES I <mark>P</mark>	78
_	MAAVR <mark>G</mark> S <mark>PPEYK</mark> VRAVEEIKRMISSK <mark>P</mark> VVAIVSFRNVPA <mark>GQ</mark> MQKIRREF <mark>RG</mark> K-AEIKVVKNTLLERALDALGGDYL	75
_	MAYKAK <mark>GQPP</mark> SCYE <mark>P</mark> KYAEWKRREYKELKELMDEYENYCLYDLEGIPAPQLQEIRAKLRERDTIIRMSRNTLMRIALEEKLDERPELE	88
_	MAHVAEWKKKEVQELHDLIK <mark>G</mark> YEVV <mark>G</mark> IANLADIPAR <mark>Q</mark> LQKMRQTLRDS-ALIRM <mark>SKKT</mark> LISLALEKA <mark>G</mark> RELENVD	74
_	MITAESEHKIA <mark>PWK</mark> IEEVNKLK <mark>E</mark> LLKN <mark>G</mark> QIVALVDMMEVPARQLQEIRDKIR-GTMTLKM <mark>SRNT</mark> LIERAIKEVAEETGNPEFA	82
_	MIDAKSEHKIA <mark>PWK</mark> IEEVNALK <mark>E</mark> LLKSANVIALIDMMEVPAVQLQEIRDKIR-DQMTLKM <mark>SRNT</mark> LIKRAVEEVAEETGNPEFA	82
_	metkakahaa <mark>pwk</mark> ieevktlk <mark>g</mark> likskpaaaiddmmdababdgfogfisuur ik-dkakikukskukseveteinekik	81
_	MAHVAEWKKKEVEELANLIKS <mark>YP</mark> VIALVDVSSMPAY <mark>P</mark> LSQMRRLIREN <mark>GGLLRVSRNT</mark> LIELAIKKAAQEL <mark>G</mark> KPELE	77
_	MAHVAEWKKKEVEELAKLIKS YP VIALVDVSSMP AYPLSQMRRLIREN <mark>GG</mark> LLRV <mark>SRNT</mark> LIELAIKKAAKEL <mark>G</mark> KPELE	77
—	MAHVAEWKKKEVEELANLIKSYPVVALVDVSSMPAYPLSQMRRLIRENNGLLRVSRNTLIELAIKKVAQELGKPELE	77
_	MAHVAEWKKKEVEELANIIKSYPVIALVDVAGVPAYPLSKMRDKL <mark>R-G</mark> KALLRV <mark>SRNT</mark> LIELAIKRAAQELGQPELE	76
_	MSAESERKTET I <mark>PEWK</mark> QEEVDAIVEMIESYESVGVVNIAGIPSRQLQDMRRDLHGT-AELRV <mark>SRNTLLERAL</mark> DDVDD <mark>GLE</mark>	79
_	MSESEVRQTEVI <mark>P</mark> QWKREEVDELVDFIESYESVGVVGVAGIPSRQLQSMRRELHGS-AAVRMSRNTLVNRALDEVNDGFE	79
_	MSAEEQRTTEEVPEWKRQEVAELVDLLETYDSVGVVNVTGIPSKQLQDMRRGLHGQ-AALRMSRNTLLVRALEEAGDGLD	79
_	MKEVSQQKKELVNEITQRIKASRSVAIVDTAGIRTRQIQDIRGKNRGK-INLKVIKKTLLFKALENLGDEKLS	72
_	MRKINPKKKEIVSELAQDITKSKAVAIVDIKGVRTRQMQDIRAKNRDK-VKIKVVKKTLLFKALDSINDEKLT	72
	MTEPAQWKIDFVKNLENE INSRKVAAIVSIKGLRNNEFQKIRNSIRDK-ARIKVSRARLLRLAIENTGKNNIV	72
ruler	$1, \ldots, 10, \ldots, 20, \ldots, 30, \ldots, 40, \ldots, 50, \ldots, 60, \ldots, 70, \ldots, 80, \ldots, 90$	

By Miguel Andrade at English Wikipedia, https://commons.wikimedia.org/w/index.php?curid=3930704

# **Multiple alignment algorithms**

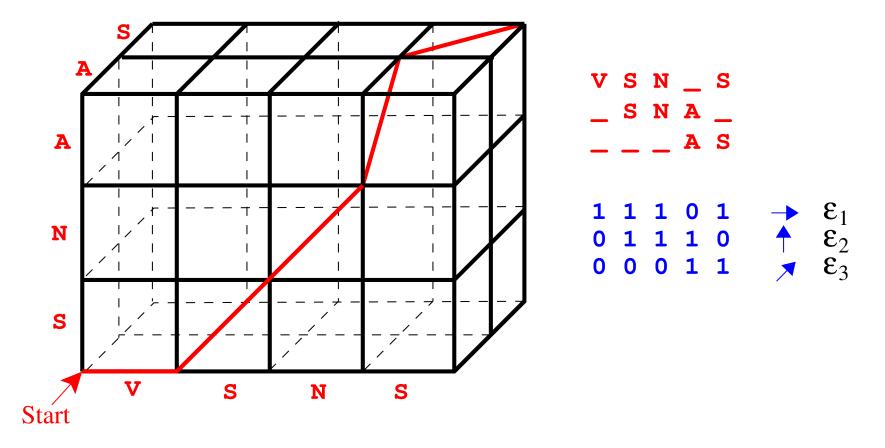
**Definition.** A multiple alignment of sequences  $X^1, \ldots, X^n$  is a series of gapped sequences  $\tilde{X}^1, \ldots, \tilde{X}^n$  such that (i)  $\tilde{X}^i$  is an extension of  $X^i$  obtained by insertions of spaces; (ii)  $|\tilde{X}^1| = |\tilde{X}^2| = \cdots = |\tilde{X}^n|$ .

Why are we interested in multiple alignments?

- A multiple alignment **carries more information** than a pairwise one, as a protein can be matched against a family of proteins instead of only against another one.
- Multiple similarity of (protein) sequences suggests
  - a common structure,
  - a common function,
  - a common evolutionary source.

# The alignment hyper-cube

Best multiple alignment of r sequences: Best path through r-dimensional hyper-cube D.



Alignment path for three example sequences.

#### **Dynamic Programming Solution**

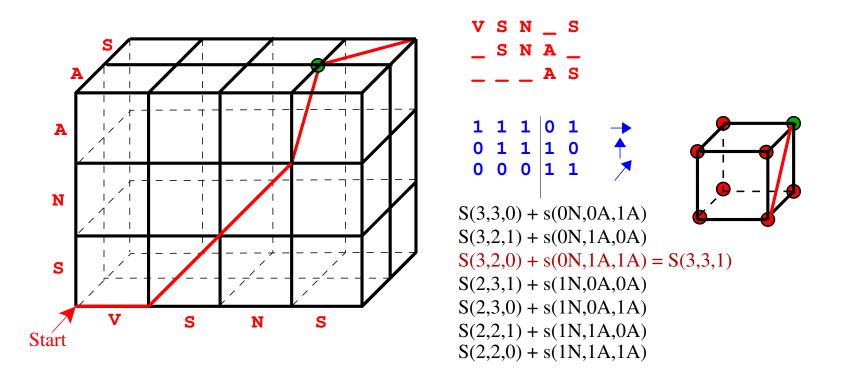
- Best multiple alignment of r sequences:
   Best path through r-dimensional hyper-cube.
- Define  $S(j_1, j_2, \ldots, j_r)$  as as the best score for aligning the prefixes of lengths  $j_1, j_2, \ldots, j_r$  of the sequences  $X^1, X^2, \ldots, X^r$ .
- We define  $S(0, 0, \dots, 0) = 0$ , and we calculate

$$S(j_1, j_2, \dots, j_r) = \max_{\substack{(\epsilon_1, \dots, \epsilon_r): \epsilon_i \in \{0, 1\}, \epsilon \neq \mathbf{0}}} \left[ S(j_1 - \epsilon_1, j_2 - \epsilon_2, \dots, j_r - \epsilon_r) + s(\epsilon_1 x_{j_1}, \dots, \epsilon_r x_{j_r}) \right],$$

where *s* is the **scoring function** (example s(a, b, 0): joint score for aligning characters *a*, *b* and a gap) and  $\epsilon = (\epsilon_1, \dots, \epsilon_r)$  is a binary vector that indicates the **directions** of the alignment progress in the hyper-cube.

#### **Dynamic Programming Solution: Complexity**

- The size of the hyper-cube is  $O(\prod_{j=1}^{r} n_j)$   $(n_j = \text{length of } x_j)$ .
- Computation of each entry considers  $2^r 1$  other entries. Example: 000, 001, 010, 011, 100, 101, 110, 111
- If  $n_1 = n_2 = \cdots = n_r = n$ , the space complexity is of  $O(n^r)$  and the time complexity is of  $O(2^r n^r)$ .



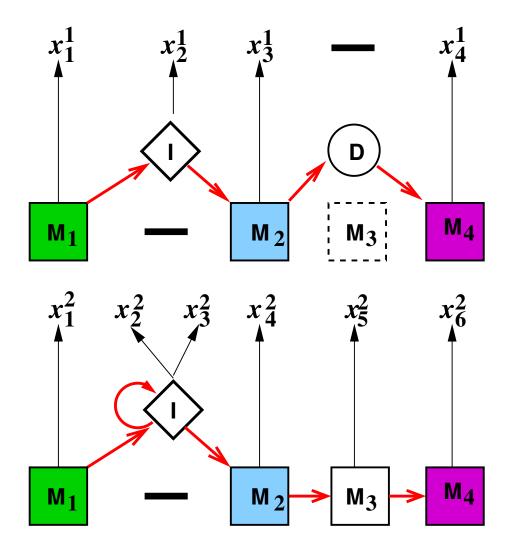
# **Scoring Metrics**

- A scoring scheme should take into account that...
  - some positions are more conserved than others
     → position specific scoring;
  - 2. the sequences are not independent, but are related by a **phylogenetic tree**.
- Ideal scoring: Complete probabilistic model of evolution
   Probability of a multiple alignment is composed of the probabilities of all evolutionary events necessary to produce the alignment.
- In practice, we do not have such a model
   *simplifying assumptions:* Two main concepts:
  - 1. Position specific, but ignoring the phylogenetic tree;
  - 2. explicit tree model, but position independent.

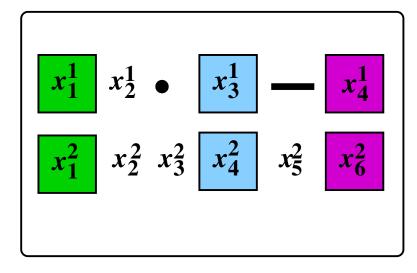
# Multiple alignments by Profile HMM training

- Suppose we have successfully trained a profile HMM from a set of labeled sequences.
   How can we use this HMM to derive the multiple alignment of n sequences?
- Answer: align all n sequences to the profile using the Viterbi algorithm ~> most probable state paths for all sequences.
- Characters aligned to the same match state are aligned in columns.
- Multiple alignments from HMMs are approximations of type one:
  - Score is **position specific**,
  - but sequences are treated as **independent objects**.

#### **Computing the multiple alignment: example**



**Multiple Alignment** 



# **Computing the multiple alignment: Real example**

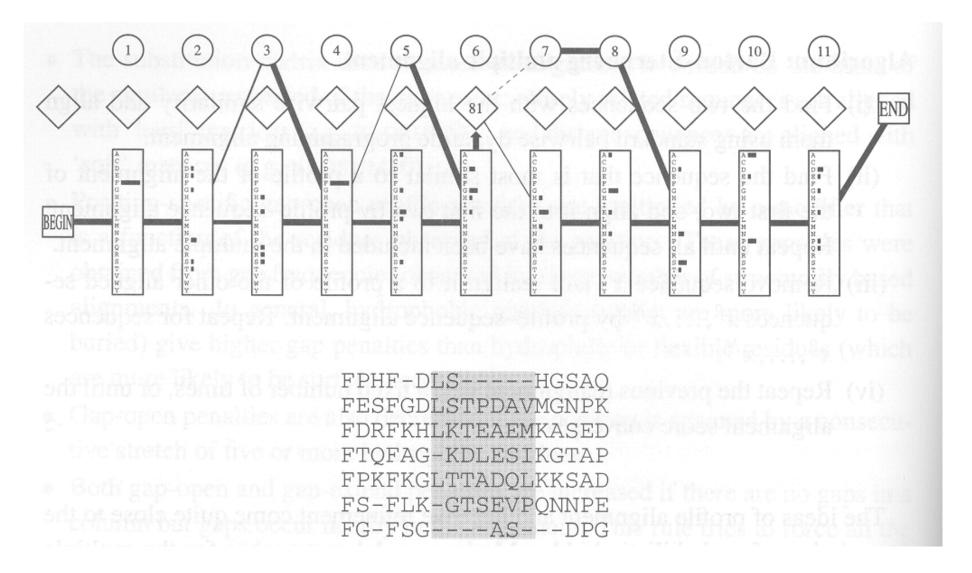


Figure 6.4 A model (top) estimated from an alignment (bottom). The characters in the shaded area of the alignment were treated as

inserts.

Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004

#### **Computing the multiple alignment: Real example**

FPHF-Dls....HGSAQ FESFGDlstpdavMGNPK FDRFKHlkteaemKASED FTQFAGkdlesi.KGTAP FPKFKGlttadqlKKSAD FS-FLKgtsevp.QNNPE FG-FSGas....-DPG FS-FLKngvdptaai--NPK FPHF-Dls....HGSAQ FESFGDlstpdav.MGNPK FDRFKHlkteaem.KASED FTQFAGkdlesi...KGTAP FPKFKGlttadql.KKSAD FS-FLKgtsevp...QNNPE FG-FSGas....-DPG

**Figure 6.6** Left: the alignment of the seven sequences is shown with lowercase letters meaning inserts. The dots are just space-filling characters to make the matches line up correctly. Right: the alignment is shown after a new sequence was added to the set. The new sequence is shown at the top, and because it has more inserts more space-filling dots were added.

Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004

# **Multiple Alignments by Profile HMM training**

- For parameter estimation in Profile HMMs, aligned training sequences are often unavailable
   usually we only have a sample of unaligned sequences, the state paths are unknown.
- Idea: Use EM algorithm for iterative parameter optimization (Baum-Welch algorithm).
- Recall: for the EM algorithm, we need the forward and backward probabilities in the E-step for calculating
  - $E_{bl}$  (the expected emission counts) and
  - $A_{l'l}$  (expected transition counts).

# **Simpler Multiple Alignment Algorithms**

- Alternative to the probabilistic HMM formulation:
   Sum of Pairs score:
   Sum of scores between all pairs of sequences.
- The SP score for a column  $m_j$  of the multiple alignment is

$$S(m_j) = \sum_{k < l} \underbrace{s(m_j^k, m_j^l)}_{\text{from scoring matrix}}$$

• SP scores lack a probabilistic justification: Correct log-odds score for 3-way alignment would be

$$s(a, b, c) = \log \frac{p_{abc}}{q_a q_b q_c} \neq \underbrace{\log \frac{p_{ab}}{q_a q_b} + \log \frac{p_{bc}}{q_b q_c} + \log \frac{p_{ac}}{q_a q_c}}_{\text{SP score}}.$$

# **Approximation Algorithms for MSA**

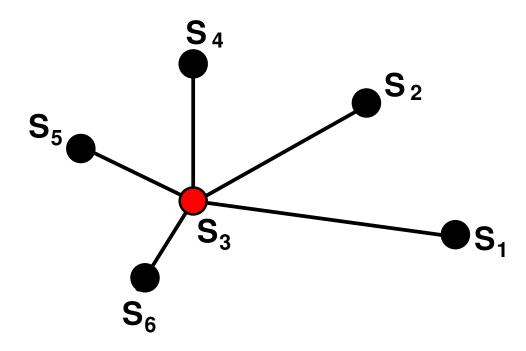
- Even for SP scores, MSA has exponential time complexity.
- Denote by D(S,T) the **minimum cost** of aligning S with T.
- Let  $\sigma(x, y)$  be our **cost function**, i.e. the cost of aligning the character x with the character y, for  $x, y \in \Sigma \cup \{-\}$ .
- Here we minimize costs  $\sigma$  instead of maximizing scores s. Example transformation:  $\sigma(x, y) = \exp(-\lambda s(x, y))$ .
- We assume that  $\sigma(-,-) = 0$ ,  $\sigma(x,y) = \sigma(y,x)$ , and that the triangle inequality holds:  $\sigma(x,y) \le \sigma(x,z) + \sigma(z,y)$

**Problem:** The SP alignment problem. **INPUT:** A set of sequences  $S = \{S_1, \ldots, S_k\}$ . **QUESTION:** Compute a global multiple alignment  $\mathcal{M}$  with minimum SP-costs, given the above assumptions on  $\sigma(\cdot, \cdot)$ .

# **The Center Star Method for Alignment**

Approximation algorithm for calculating the optimal multiple alignment under the SP metric with **approximation ratio of two.** 

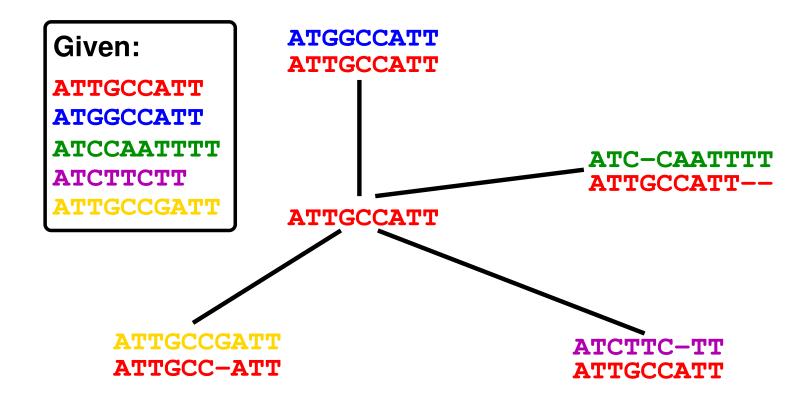
- Center string: String that minimizes  $\sum_{S_j \in S} D(S_c, S_j)$ .
- Center star: A star tree of k nodes, center node labeled S<sub>c</sub>, each of the k − 1 remaining nodes labeled by S \ {S<sub>c</sub>}.



**Type-2 approximation:** explicit (star-)tree model, but position independent scoring.

# **The Center Star Algorithm**

1. Find  $S_t \in S$  minimizing  $\sum_{i \neq t} D(S_i, S_t)$  and let  $\mathcal{M} = \{S_t\}$ 



2. Add sequences in S \ {S<sub>t</sub>} to M one by one so that the pairwise alignment of every newly added sequence with S<sub>t</sub> is optimal.
Add spaces, when needed, to all pre-aligned sequences.

#### **The Center Star Algorithm**

Given:
ATTGCCATT
ATGGCCATT
ATCCAATTTT
ATCTTCTT
ATTGCCGATT

ATGGCCATT ATTGCCATT

Pair:

ATC-CAATTTT ATTGCCATT--

ATCTTC-TT ATTGCCATT

ATTGCCGATT ATTGCC-ATT Alignment: ATTGCCATT ATGGCCATT

ATTGCCATT--ATGGCCATT--ATC-CAATTTT

ATTGCCATT--ATGGCCATT--ATC-CAATTTT ATCTTC-TT--

ATTGCC-ATT--ATGGCC-ATT--ATC-CA-ATTTT ATCTTC--TT--ATTGCCGATT--

# The Center Star Algorithm: Analysis

- $\mathcal{M}$  : Multiple alignment produced by the center-star algorithm.
- d(i,j): Cost of the resulting pairwise alignment of  $S_i$  and  $S_j$ , induced by  $\mathcal{M}$ .

Note that  $\underline{D(S_i, S_j)} \leq \underline{d(i, j)}$ 

cost of best pairwise alignment cost of induced alignment

- SP-costs of center-star alignment:  $\sigma(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j=1, j \neq i}^{k} d(i, j)$
- $\mathcal{M}^*$ : Optimal SP-alignment of all strings in  $\mathcal{S}$  with costs  $\sigma(\mathcal{M}^*)$ .

#### Theorem 1.

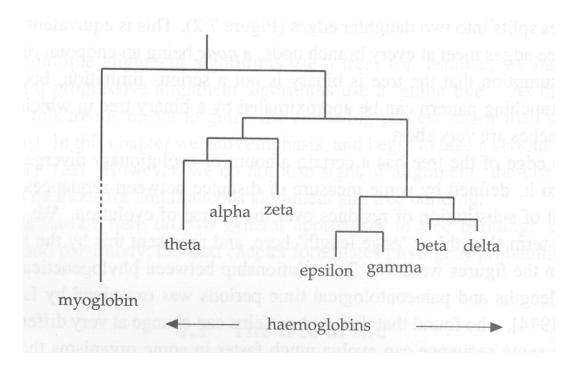
$$\frac{\sigma(\mathcal{M})}{\sigma(\mathcal{M}^*)} = \leq \frac{2(k-1)}{k} \leq 2.$$

**Theorem 2.** The running time of the center star algorithm for kstrings with length  $\leq n$  is  $O(k^2 \cdot n^2)$ .

Proofs: see exercises.

#### **Progressive alignment heuristics**

Idea: Use a binary "guide tree" instead of a star tree (Guide tree defines a model of evolution) Leaves: sequences, inner nodes: alignments (sequence-sequence, sequence-profile, or profile-profile).



Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004

# **Progressive alignment: ClustalW**

ClustalW is a software package for multiple alignment (implementing an algorithm of Thompson, Higgins, Gibson 1994).

- 1. Calculate all pairwise alignment scores, convert to **pairwise distances.**
- 2. Use **Neighbor-Joining** algorithm to build a **tree** from the distances.
- 3. Align sequence sequence, sequence profile, profile profile.

This algorithm makes use of many ad-hoc rules such as weighting, different matrix scores and special gap scores.

