Bioinformatics 1

Biology, Sequences, Phylogenetics

Part 3

Sepp Hochreiter

Contents

4 Multiple Alignment
4.1 Motivation
4.2 Multiple Sequence Similarities and Scoring
4.2.1 Consensus and Entropy Score
4.2.2 Tree and Star Score
4.2.3 Weighted Sum of Pairs Score
4.3 Multiple Alignment Algorithms
4.3.1 Exact Methods
4.3.2 Progressive Algorithms
4.3.3 Other Multiple Alignment Algorithms
4.4 Profiles and Position Specific Scoring Matrices

Motivation

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

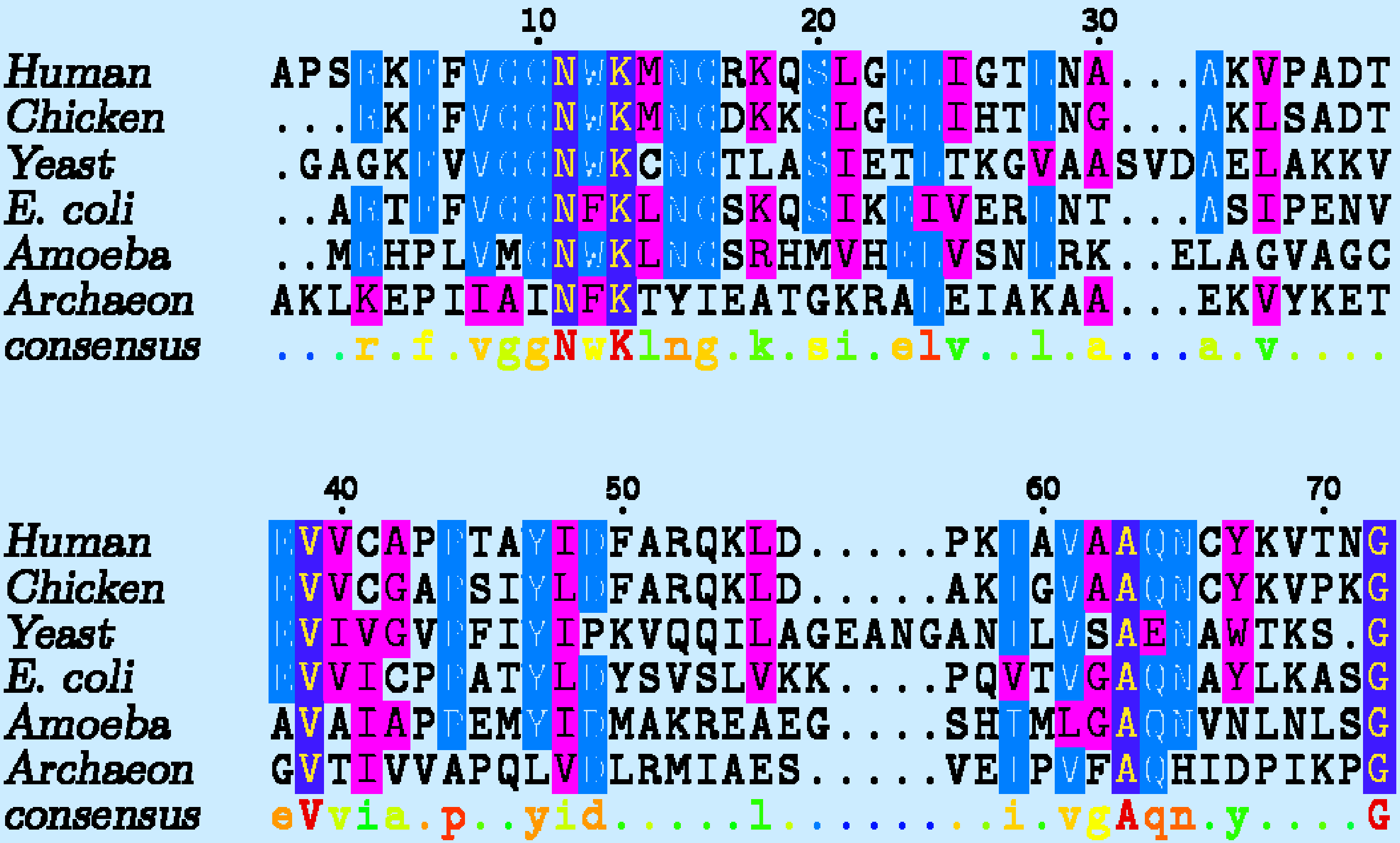
4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

Compare more than two sequences: arranged sequences so that the amino acids for every the columns match as good as possible

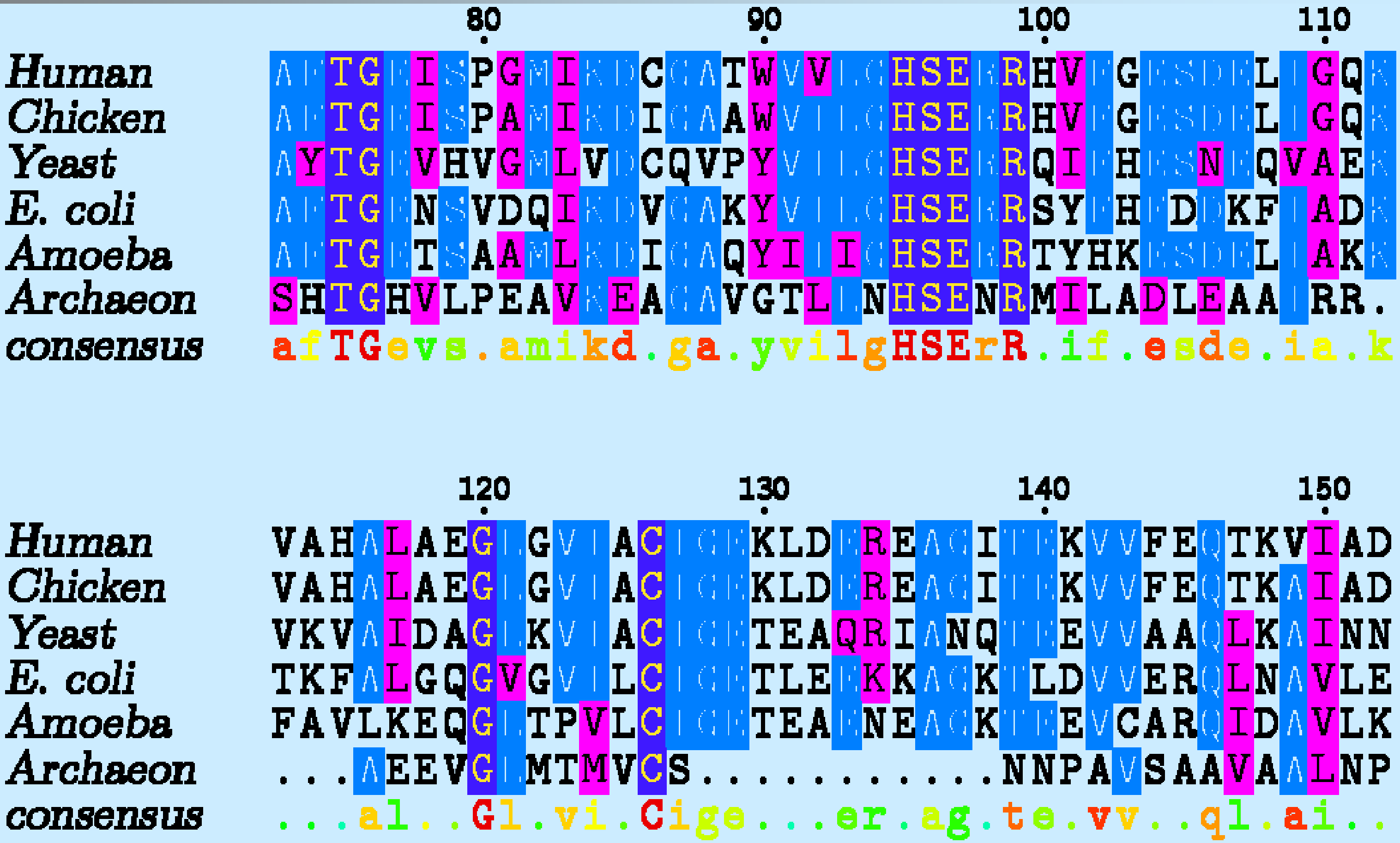


```
                   10        20        30
Human      APSRKFFVGGNWKMNGRKQSLGELIGTLNA...AKVPADT
Chicken    ...RKFFVGGNWKMNGDKKSLGELIHTLNG...AKLSADT
Yeast      .GAGKFVVGGNWKCNGTLASIETLTKGVAASVDAELAKKV
E. coli    ..ARTFFVGGNFKLNGSKQSIKEIVERLNT...ASIPENV
Amoeba     ..MRHPLVMGNWKLNGSRHMVHELVSNLRK..ELAGVAGC
Archaeon   AKLKEPIIAINFKTYIEATGKRALEIAKAA...EKVYKET
consensus  ...r.f.vggNwKlng.k.si.elv..l.a...a.v....

             40        50             60        70
Human      EVVCAPPTAYIDFARQKLD.....PKIAVAAQNCYKVTNG
Chicken    EVVCGAPSIYLDFARQKLD.....AKIGVAAQNCYKVPKG
Yeast      EVIVGVPFIYIPKVQQILAGEANGANILVSAENAWTKS.G
E. coli    EVVICPPATYLDYSVSLVKK....PQVTVGAQNAYLKASG
Amoeba     AVAIAPTEMYIDMAKREAEG....SHTMLGAQNVNLNLSG
Archaeon   GVTIVVAPQLVDLRMIAES.....VEIPVFAQHIDPIKPG
consensus  eVvia.p..yid.....l........i.vgAqn.y....G
```

Motivation

- 4 Multiple Alignment
- 4.1 Motivation
- 4.2 Scoring
- 4.2.1 Consensus
- 4.2.2 Tree and Star
- 4.2.3 Sum of Pairs
- 4.3 Algorithms
- 4.3.1 Exact Methods
- 4.3.2 Progressive
- 4.3.3 Other
- 4.4 Profiles / PSSMs



```
                 80        90       100       110
                  .         .         .         .
Human      AFTGEISPGMIKDCGATWVVLGHSEKRHVFGESDELIGQK
Chicken    AFTGEISPAMIKDIGAAWVILGHSEKRHVFGESDELIGQK
Yeast      AYTGEVHVGMLVDCQVPYVILGHSEKRQIFHESNEQVAEK
E. coli    AFTGENSVDQIKDVGAKYVILGHSEKRSYFHFDDKFIADK
Amoeba     AFTGETSAAMLKDIGAQYIIIGHSEKRTYHKESDELIAKK
Archaeon   SHTGHVLPEAVKEAGAVGTLLNHSENRMILADLEAAIRR.
consensus  afTGevs.amikd.ga.yvilgHSErR.if.esde.ia.k

                120       130       140       150
                  .         .         .         .
Human      VAHALAEGLGVIACIGEKLDEREAGITEKVVFEQTKVIAD
Chicken    VAHALAEGLGVIACIGEKLDEREAGITEKVVFEQTKAIAD
Yeast      VKVAIDAGLKVIACIGETEAQRIANQTEEVVAAQLKAINN
E. coli    TKFALGQGVGVILCIGETLEEKKAGKTLDVVERQLNAVLE
Amoeba     FAVLKEQGLTPVLCIGETEAENEAGKTEEVCARQIDAVLK
Archaeon   ...AEEVGLMTMVCS...........NNPAVSAAVAALNP
consensus  ...al..Gl.vi.Cige...er.ag.te.vv..ql.ai..
```

Motivation

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

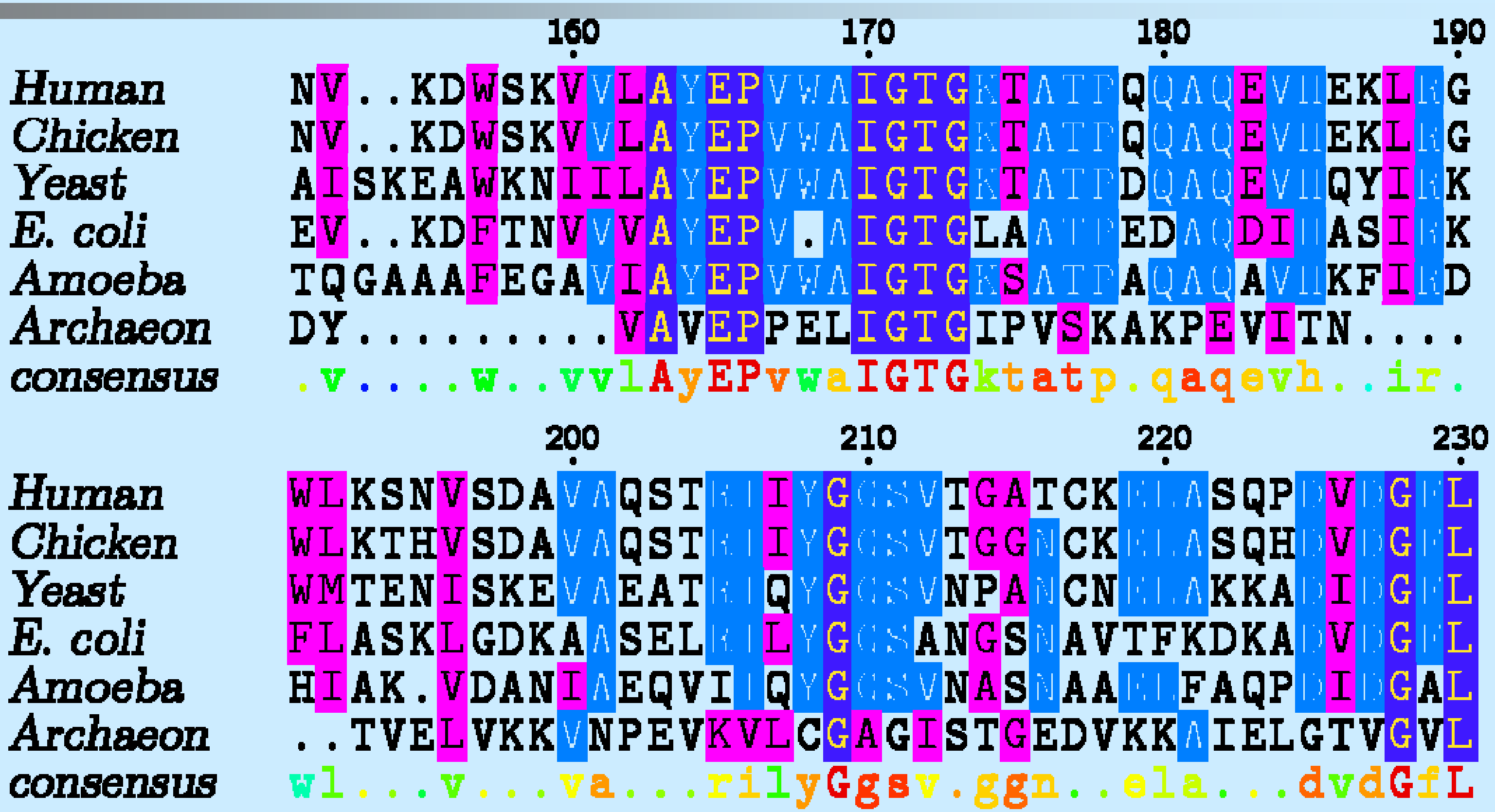
4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

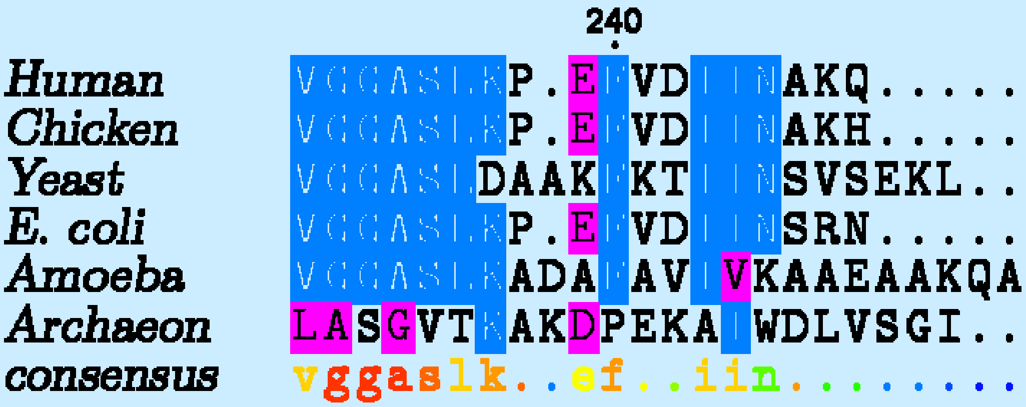
4.4 Profiles / PSSMs



```
                   160       170       180       190
                     .         .         .         .
Human      NV..KDWSKVVLAYEPVWAIGTGKTATPQQAQEVHEKLRG
Chicken    NV..KDWSKVVLAYEPVWAIGTGKTATPQQAQEVHEKLRG
Yeast      AISKEAWKNIILAYEPVWAIGTGKTATPDQAQEVHQYIRK
E. coli    EV..KDFTNVVVAYEPV.AIGTGLAATPEDAQDIHASIRK
Amoeba     TQGAAAFEGAVIAYEPVWAIGTGKSATPAQAQAVHKFIRD
Archaeon   DY.........VAVEPPELIGTGIPVSKAKPEVITN....
consensus  .v....w..vvlAyEPvwaIGTGktatp.qaqevh..ir.

                   200       210       220       230
                     .         .         .         .
Human      WLKSNVSDAVAQSTRIIYGGSVTGATCKELASQPDVDGFL
Chicken    WLKTHVSDAVAQSTRIIYGGSVTGGNCKELASQHDVDGFL
Yeast      WMTENISKEVAEATRIQYGGSVNPANCNELAKKADIDGFL
E. coli    FLASKLGDKAASELRILYGGSANGSNAVTFKDKADVDGFL
Amoeba     HIAK.VDANIAEQVIIQYGGSVNASNAAELFAQPDIDGAL
Archaeon   ..TVELVKKVNPEVKVLCGAGISTGEDVKKAIELGTVGVL
consensus  wl...v...va...rilyGgsv.ggn..ela...dvdGfL
```

Motivation



```
Human      VGGASLKP.EFVDIINAKQ.....
Chicken    VGGASLKP.EFVDIINAKH.....
Yeast      VGGASLDAAKFKTIINSVSEKL..
E. coli    VGGASLKP.EFVDIINSRN.....
Amoeba     VGGASLKADAFAVIVKAAEAAKQA
Archaeon   LASGVTKAKDPEKAIWDLVSGI..
consensus  vggaslk..ef..iin........
```

Motivation

Multiple sequence alignment is used to

- detect remote homologous regions
- detect motifs (regular patterns) in protein families
- detect conserved regions or positions (disulfide bonds)
- detect structural blocks like helices or sheets
- construct phylogenetic trees
- construct a profiles (search or averages)
- sequence genomes by superimposing fragments (nucleotides)
- cluster proteins according to similar regions

Scoring and Similarity

Similarity measures can be based on:

- ➥ the similarity of all sequences to a reference sequence
- ➥ the similarities between evolutionary adjacent sequences
- ➥ all pairwise similarities

Consensus and Entropy

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

consensus sequence: obtained if for each column in the alignment the most frequent amino acid is chosen
more precisely: the amino acid or letter which has the highest score to all other amino acids or gaps in the column

consensus score: sum of the pairwise score between sequences and the consensus sequence

generalized by profiles instead of sequences

profile: relative frequency instead of most frequent

Consensus and Entropy

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

high entropy of the letter distribution: all letter are equally probable
zero entropy: one letter in the column

good alignment correlates with a low accumulative entropy

entropy score:

$$-\sum_{i}\sum_{a} f_{i,a} \, \log f_{i,a}$$

$f_{i,a}$: relative frequency of letter a in column i

Tree and Star Score

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

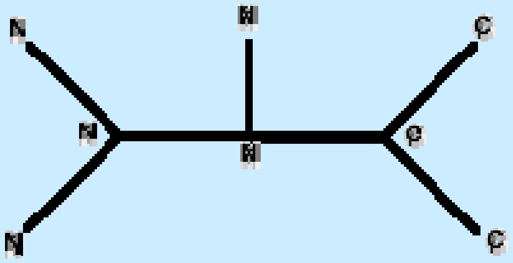
To count the number of mutations only those pairs should be compared which are evolutionary adjacent

E
E
E
E
D
D
D
D

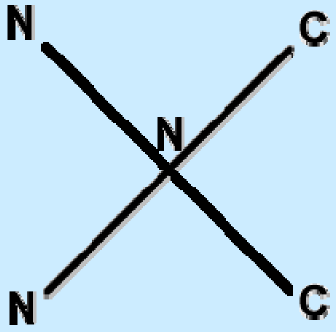
evolutionary adjacent sequences are represented through a phylogenetic tree, which must be constructed

Tree and Star Score

```
NNN
NNN
NNN
NNC
NCC
```



phylogenetic star: one sequence is considered as ancestor



Weighted Sum of Pairs

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

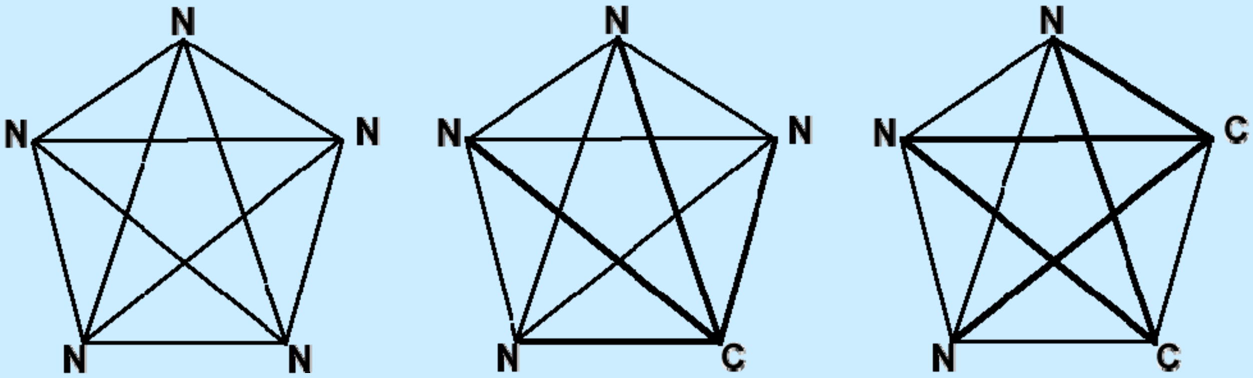
4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

weighted sum of pairs: all pairwise comparisons



alignment length: L
number sequences: N

$$\sum_{i=1}^{L} \sum_{l=1}^{N-1} \sum_{j=l+1}^{N} w_{l,j} \; s\left(x_{i,l}, x_{i,j}\right)$$

weights: reduce the influence of closely related sequences

Weighted Sum of Pairs

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

Disadvantage: relatively decreases with respect of N for conservative regions; but larger N means more conservative

$$S_{\text{old}} = \frac{N\ (N-1)}{2} s(C,C)$$

N Cs vs. (N-1) Cs and D

$$S_{\text{new}} = \frac{N\ (N-1)}{2} s(C,C) - (N-1)s(C,C) + (N-1)s(C,D)$$

$$\frac{S_{\text{old}} - S_{\text{new}}}{S_{\text{old}}} = \frac{2\ (N-1)\ s(C,C) - 2\ (N-1)\ s(C,D)}{N\ (N-1)\ s(C,C)} =$$

$$\frac{2}{N} \left(1 - \frac{s(C,D)}{s(C,C)} \right)$$

for large N small difference

$$s(C,D) < s(C,C)$$

reasonable scoring matrices: $\left(1 - \frac{s(C,D)}{s(C,C)} \right) > 0$

Weighted Sum of Pairs

contra-intuitive: a new letter in a column of 100 equal letters is more surprising as a new letter in a column of 3 equal letters

Information gain: $-\log f_{i,a} \;=\; \log(N)$

Gaps: as for pairwise algorithms, linear gaps more efficient

Multiple Alignment Algorithms

multiple alignment optimization problem: NP-hard

Exact solution: only 10 to 15 sequences

algorithm classes:

➥ global and progressive methods: MSA, COSA, GSA, clustalW, TCoffee

➥ iterative and search algorithms: DIALIGN, MultAlin, SAGA, PRRP, Realigner

➥ local methods (motif/profile): eMotif, Blocks, Dialign, Prosite, HMM, Gibbs sampling

➥ divide-and-conquer algorithms: DCA, OMA

Multiple Alignment Algorithms

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

	Global progressive alignments methods	
CLUSTALW	ftp://ftp.ebi.ac.uk/pub/software	Thompson et al. (1994/97) Higgins et al. (1996)
MSA	http://www.psc.edu/ http://www.ibc.wustl.edu/ibc/msa.html ftp://fastlink.nih.gov/pub/msa	Lipman et al. (1989) Gupta et al. (1995)
PRALINE	http://mathbio.nimr.mrc.ac.uk/~jhering/praline	Heringa (1999)
	Iterative and search algorithms	
DIALIGN segment alignment	http://www.gsf.de/biodv/dialign.html	Morgenstern et al. (1996)
MultAlin	http://protein.toulouse.inra.fr/multalin.html	Corpet (1988)
PRRP progressive global alignment	ftp://ftp.genome.ad.jp/pub/genome/saitamacc	Gotoh (1996)
SAGA genetic algorithm	http://igs-server.cnrs-mrs.fr/~cnotred/Projects_home_page/saga_home_page.html	Notredame and Higgins (1996)

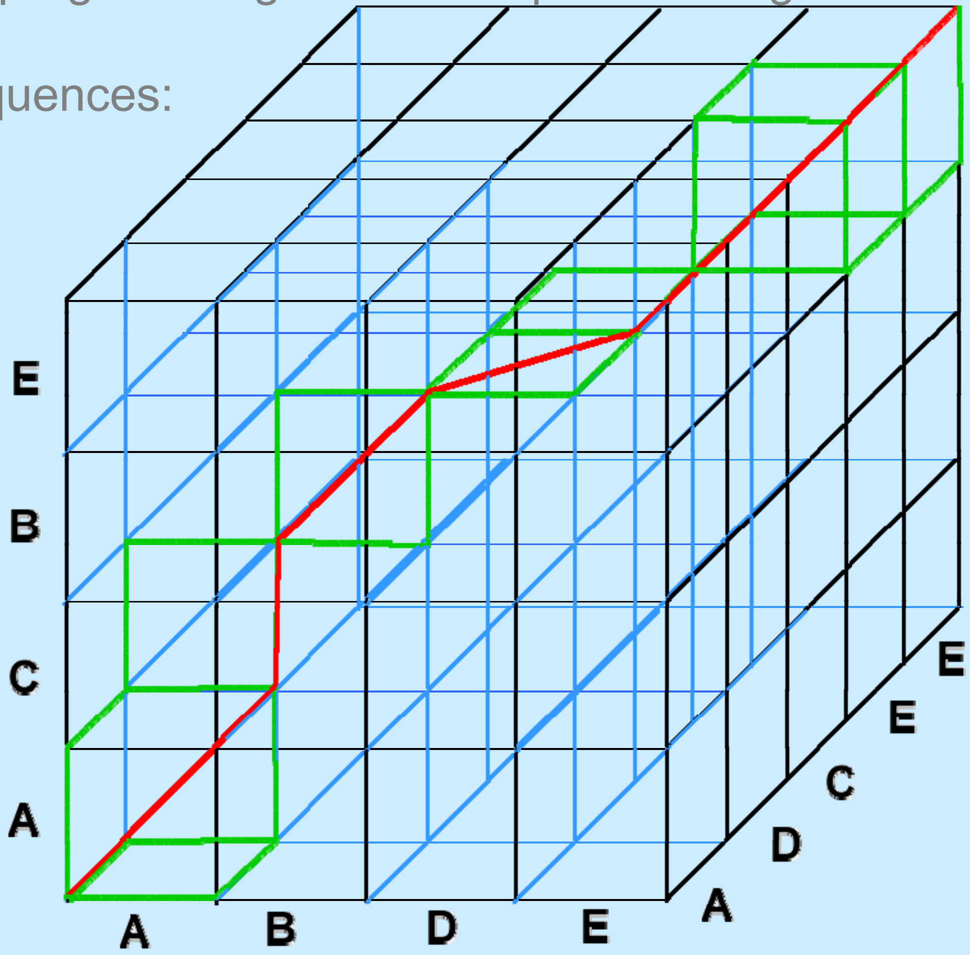
	Local alignments / motif / profile	
Aligned Segment Statistical Eval. Tool (Asset)	ftp://ncbi.nlm.nih.gov/pub/neuwald/asset	Neuwald and Green (1994)
BLOCKS	http://blocks.fhcrc.org/blocks/	Henikoff and Henikoff (1991, 1992)
eMOTIF	http://dna.Stanford.EDU/emotif/	Nevill-Manning et al. (1998)
GIBBS (Gibbs sampler)	ftp://ncbi.nlm.nih.gov/pub/neuwald/gibbs9_95/	Lawrence et al. (1993) Liu et al. (1995) Neuwald et al. (1995)
HMMER hidden Markov model	http://hmmer.wustl.edu/	Eddy (1998)
MACAW	ftp://ncbi.nlm.nih.gov/pub/macaw	Schuler et al. (1991)
MEME (EM method)	http://meme.sdsc.edu/meme/website/	Bailey and Elkan (1995) Grundy et al. (1996, 1997) Bailey and Gribskov (1998)
Profile (UCSD)	http://www.sdsc.edu/projects/profile/	Gribskov and Veretnik (1996)
SAM hidden Markov model	http://www.cse.ucsc.edu/research/comp/bio/sam.html	Krogh et al. (1994) Hughey and Krogh (1996)

Exact Methods

4 Multiple Alignment
4.1 Motivation
4.2 Scoring
4.2.1 Consensus
4.2.2 Tree and Star
4.2.3 Sum of Pairs
4.3 Algorithms
4.3.1 Exact Methods
4.3.2 Progressive
4.3.3 Other
4.4 Profiles / PSSMs

MSA (Lippman et al., 1989, Gupa et al., 1995): generalizes the dynamic programming ideas from pairwise alignment

three sequences:



```
--E
EEE
--C
D-D
BB-
-C-
AAA
```

```
A-BD-E-
ACB--E-
A--DCEE
```

Exact Methods

memory and computational complexity: exponentially with N

Gupa et al., 1995: pairwise alignments constrain the path and not the whole hypercube must be filled

MSA (Gupa):

1. compute all pairwise alignment scores $S_{k,l}$
2. predict a phylogenetic tree based on the pairwise scores
3. compute pairwise weights based on the tree
4. construct a temporary multiple alignment with score S_t
5. Compute $B_{k,l}$ a lower bound on $S[k,l]$ the score of the projection of the optimal multiple alignment to k and l
6. Compute space constraints similar to the Baum-Welch
7. compute the optimal alignment on the constraint cube; Dijkstra's shortest path algorithm for nonnegative edges; priority queue; non-negativity guarantees monotone increasing costs
8. compare the weight in the alignment with the maximal weight

Exact Methods

last step compares actual and maximal weight, if actual is larger then a better alignment may be possible, larger maximal weight means more computational costs

$$\boxed{B_{k,l} \; = \; S_t \; + \; S_{k,l} \; - \; \sum_{i,j} S_{i,j}}$$

Carillo-Lipman bound:

$$S \geq S_t$$

$$\Leftrightarrow \quad \sum_{i,j} S[i,j] \geq S_t$$

$$\Rightarrow \quad \sum_{(i,j)\neq(k,l)} S_{i,j} \; + \; S[k,l] \geq S_t$$

$$\boxed{\begin{array}{l} S[k,l] \;\leq\; S_{k,l} \\ S_t \;\leq\; S \end{array}}$$

$$\Leftrightarrow \quad S[k,l] \geq \; S_t \; - \sum_{(i,j)\neq(k,l)} S_{i,j}$$

$$\Leftrightarrow \quad S[k,l] \; \geq \; S_t \; + \; S_{k,l} \; - \; \sum_{i,j} S_{i,j}$$

$$\Leftrightarrow \quad S[k,l] \; \geq \; B_{k,l}$$

Exact Methods

4 Multiple Alignment
4.1 Motivation
4.2 Scoring
4.2.1 Consensus
4.2.2 Tree and Star
4.2.3 Sum of Pairs
4.3 Algorithms
4.3.1 Exact Methods
4.3.2 Progressive
4.3.3 Other
4.4 Profiles / PSSMs

MSA improved by the $\mathcal{A}^*$ algorithm (Lermen and Reinert, 1997)

Algorithm 1 A^*-algorithm.

Input: graph (the graph), start (start node), goal (goal node), h(s) approximation of the distance of node s to the goal, S (priority queue), N (list of visited nodes)
Output: list P of the shortest path

```
BEGIN FUNCTION
  insert (start,S)
  while not isEmpty(S) do
    current_node = pop(S)
    if current_node in N then {no path from start to goal}
      return "no path"
    end if
    insert (current_node, N)
    if current_node = goal then
      reconstruct_shortest_path(start,goal, graph)
    else {find all nodes accessible from current node}
      successors = expand(current_node, graph)
      save_predecessor_in_graph(current_node, graph)
      for all s in successors do {save node which lead to s}
        predecessor(s) = current_node {compute and store costs}
        cost(s) = cost(current_node) + edge(graph,current_node,s)
        all_cost(s) = cost(s) + h(s)
        insert(s,S) {according to all_cost(s)}
      end for
    end if
  end while
  return "no path found"
END FUNCTION
```

```
BEGIN SUBFUNCTION {shortest path P as list}
 reconstruct_shortest_path (start, node, graph)
  if node not= start then
    push(node, P) {get predecessor}
    predecessor = getPredecessor(node, graph)
    reconstruct_shortest_path (start, predecessor, graph)
  else
    return P
  end if
END SUBFUNCTION
```

Exact Methods

MSA: weighted sum of pairs and a linear gap penalty

Weight: difference pairwise and projected multiple alignment (larger difference means higher weight)

similar sequences: pull the multiple alignment towards them which down-weights them

weights through the phylogenetic tree remove weights between distant sequences

Summing up all the weights: overall divergence of the sequences

Progressive Methods

Progressive methods are the most popular methods for multiple alignment: ClustalW (Thomson,Higgins,Gibson, 1994) and TCoffee (Notredame, Higgins, Heringa, 2000)

ClustalW and TCoffee:
➥ perform pairwise alignment for each pair
➥ weight matrix: one minus the ratio of perfect matches
➥ construct a phylogenetic tree (Neighbor-Joining method)
➥ alignments between pairs sequences/alignments (start with closest distance); alignments are propagated through the tree

Initial alignments may be found through local alignment

phylogenetic tree supplies the weighting factors

Progressive Methods

Disadavantage progressive methods:
↪ local minima
↪ same scoring matrix for close and remote related sequences and same gap parameters

ClustalW

gap penalties context dependent:
↪ gaps in hydrophobic regions are more penalized
↪ gaps which are within eight amino acids to other gaps are more penalized
↪ gaps in regions of other gaps have lower gap opening penalty
↪ gap penalties are amino acid dependent

Progressive Methods

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

scoring matrices are adapted:

↪ scoring matrix from the PAM or the BLOSUM families

sequences are weighted through a phylogenetic tree:

↪ similar sequences lower weights (unbalanced data sets)

↪ phylogentic tree weights with w_i as the weight of sequence i

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_i \; w_j \; s(i,j)$$

adaptive phylogenetic tree:

↪ insufficient scores change the tree

initial gap penalty parameters:

↪ according to scoring matrix

↪ similarity of the sequences (% identity)

↪ length of the sequences (log of the shorter sequences is added)

↪ difference of the length to avoid gaps in the shorter sequence

$$\cdot \; (1 \; + \; |\log(n/m)|)$$

Progressive Methods

TCoffee (Tree based Consistency Objective Function For alignmEnt Evaluation) often better alignment than clustalW

TCoffee work as follows:

↪ libraries of pairwise aligments based on both global (clustalW) and local (FASTA) alignments (combination is more reliable)

↪ library weights are computed according to % identity

↪ libraries are combined and extended; arithmetic mean of weights; extension by aligning two sequences through a third sequence

↪ progressive alignment with a distance based on extended library

Other Methods

Center Star Alignment

center sequence $\bar{i}$: $\bar{i} \; = \; \arg\min_i \sum_j C(i,j)$

pairwise alignment costs $C(i,j)$

$\bar{i} \; = \; 1$

new sequence is added to the set of aligned seuqences by
a pairwise alignment to the center sequence introducing new gaps

Other Methods

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

Gusfield, 1993: cost is less then twice as of the optimal cost, if

$$C(i,i) = 0 \quad \text{and} \quad C(i,j) \;\leq\; C(i,k) \;+\; C(k,j)$$

scoring matrix s with

$$s(-,-) \;=\; 0$$

$$s(-,i) \;<\; 0$$

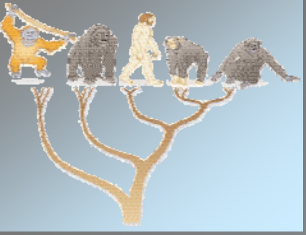
$$s(k,k) \;\geq\; s(i,k) \;+\; s(k,j) \;-\; s(i,j)$$

```
A B      A B
| |  >   | |
A C      C A
```

Then $C(i,j) \;=\; S_{i,i} \;-\; 2\;S_{i,j} \;+\; S_{j,j}$ fulfills above conditions

The second conditions is

$$S_{i,i} \;-\; 2\;S_{i,j} \;+\; S_{j,j} \;\leq\; S_{i,i} \;-\; 2\;S_{i,k} \;+\; S_{k,k} \;+$$
$$S_{k,k} \;-\; 2\;S_{k,j} \;+\; S_{j,j}$$
$$\Leftrightarrow \quad S_{i,j} \;\geq\; S_{i,k} \;+\; S_{k,j} \;-\; S_{k,k}$$



Other Methods

align i to k and j to k then align i, j, and k based on the pairwise alignments, the alignment has a gap if a gap was in one alignment

S is score of the multiple alignment

Per construction: $S[i,k] = S_{i,k}$, $S[k,j] = S_{k,j}$ and $S[k,k] = S_{k,k}$

Componentwise holds: $s(i,j) \ \geq \ s(i,k) \ + \ s(k,j) \ - \ s(k,k)$

Therefore $S[i,j] \ \geq \ S[i,k] \ + \ S[k,j] \ - \ S[k,k]$ and

$$S[i,j] \ \geq \ S_{i,k} \ + \ S_{k,j} \ - \ S_{k,k}$$

inequality to show follows from $S_{i,j} \ \geq \ S[i,j]$

Other Methods

idea of the proof of Gusfield center sequence alignment with cost C and the optimal cost C^*

$$C = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} C(i,j) \leq$$

$$\sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} C(i,1) + C(1,j) = 2\,(N-1) \sum_{i=2}^{N} C(i,1)$$

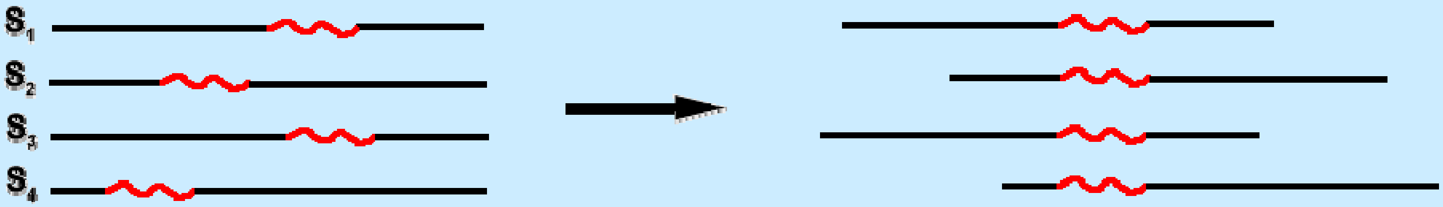
$$C^* = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} C(i,j) \geq$$

$$\sum_{i=1}^{N} \sum_{j=2}^{N} C(i,1) = N \sum_{i=2}^{N} C(i,1)$$

$$\Rightarrow \frac{C}{C^*} \leq \frac{2(N-1)}{N} \leq 2$$

Other Methods

Motifs or pattern can be superimposed for alignment landmarks



Profiles and blocks can be derived from multiple alignments

Other Methods

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

4.3.1 Exact Methods

4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

SAGA (Sequence Alignment by Genetic Algorithm): genetic algorithm

MSASA (Multiple Sequence Alignment by Simulated Annealing): simulated annealing

Gibbs sampling

HMMs (hidden Markov models) can be used to find motifs

Other Methods

4 Multiple Alignment

4.1 Motivation

4.2 Scoring

4.2.1 Consensus

4.2.2 Tree and Star

4.2.3 Sum of Pairs

4.3 Algorithms

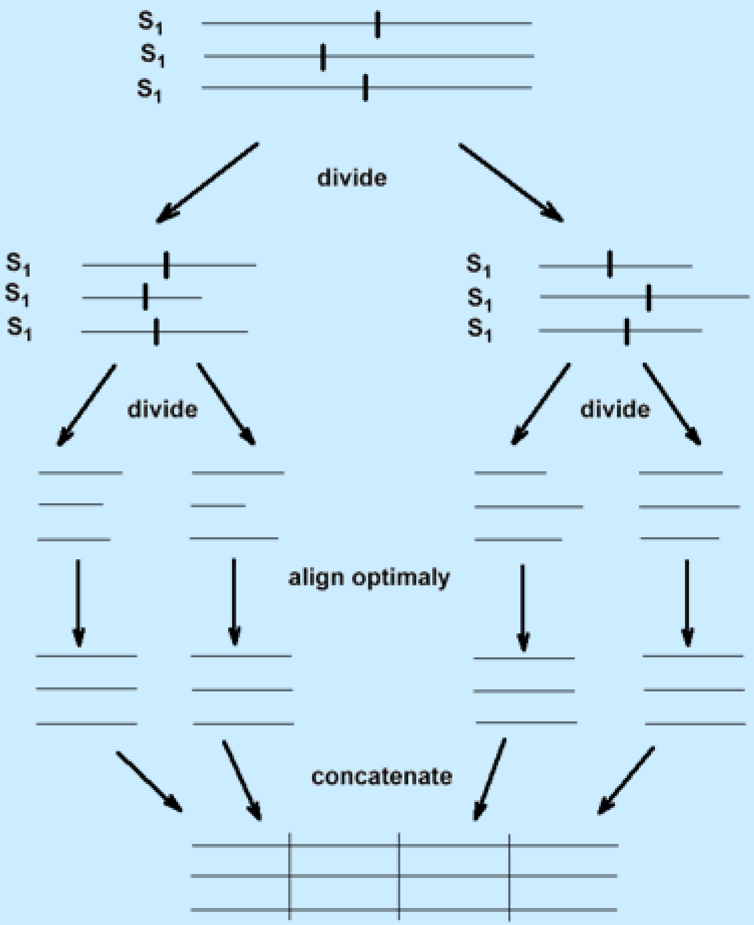
4.3.1 Exact Methods

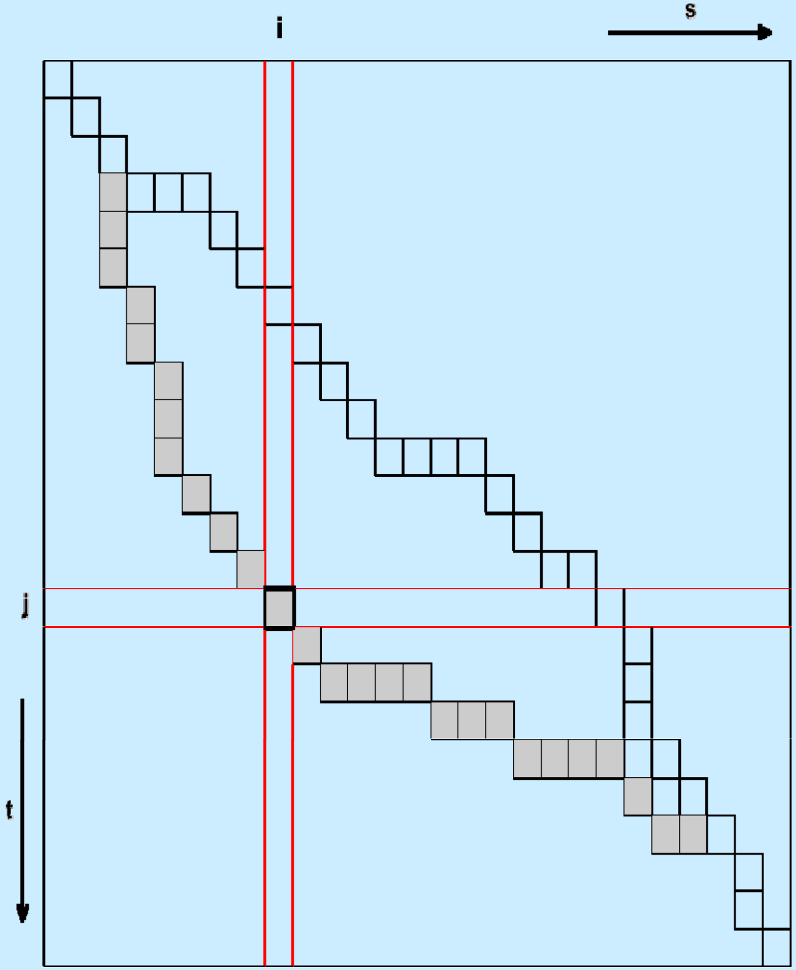
4.3.2 Progressive

4.3.3 Other

4.4 Profiles / PSSMs

Divide-and-conquer Algorithms





Profiles and PSSMs

Profiles and Position Specific Scoring Matrices

Assumptions:

- ➥ $\boldsymbol{x}$ is i.i.d. in its elements according to p_x
- ➥ n the length of $\boldsymbol{x}$ is large
- ➥ expected letter score for random sequences $\sum_i \; p_x(i) \; s(i) \; < \; 0$
- ➥ exist i for which $s(i) \; > \; 0$

$$S_n \; = \; \sum_{i=1}^{n} \; s(i)$$

centered value: $\tilde{S}_n \; = \; S_n \; - \; \frac{\ln n}{\lambda}$

$$P\left(\tilde{S}_n > S\right) \; \approx \; 1 - \exp\left(-K \; e^{-\lambda \; S}\right) \; \approx \; K \; e^{-\lambda \; S}$$

$$\sum_i p_x(i) \; \exp(\lambda \; s(i)) \; = \; 1$$

Profiles and PSSMs

q_i : frequency of a letter a_i in a column of a multiple alignment

for sufficient high scoring segments

$$\lim_{n \to \infty} q_i \;=\; p_x(i)\;\exp(\lambda\; s(i))$$

$$\Rightarrow \quad s(i) \;=\; \ln\left(\frac{q_i}{p_x(i)}\right)/\lambda$$

“Position Specific Scoring Matrices” (PSSMs) or profiles

new sequence: high scores mean similar alignment sequences