Lecture 2: Profile HMMs for sequence families

-Profile HMM
-Learning profile HMMs
-Multiple alignment and profile HMMs
-(0ther multiple alignment methods)

Profile HMMs

- Models for (amino acid) *sequence families*
- Special *structure* (match, insert, and delete states; specific transition structure)
- Parameter estimation from given *multiple alignment*
- Can be used for examining the relation of new sequences to the family represented by the profile
- So-called motifs (PWMs, PSSMs) are a special case

Multiple Alignment of Globins

Fig. 1 Alignment of 9 representative globins and *Sperm whale* (SW) myoglobin. Eight alpha helices are shown as a-h above the alignment. Numbers between brackets indicate the number of amino acids preceding and following the globin domain. [Hoogewijs *et al. BMC Genomics* 2007 **8**:356]

Ungapped score matrices

- For example, helix a in Fig. 1 is ungapped (16 columns)
- Associate with each (ungapped) position (column) i = 1, ..., L a probability distribution of symbols: $e_i(a)$ = position i has symbol a ε Σ with probability $e_i(a)$
- ML estimate:
  $e_i(a)$ = #a / #aligned_sequences
- Hence, assuming independence of positions (Bernoulli!)
  - the probability of sequence x is: $P(x) = \prod_i e_i(x_i)$
  - the log-odds with respect to random model (q_a) (= the background) is $S(x) = \sum_i \log(e_i(x_i)/q_{x(i)})$
- The resulting score matrix $(e_i(a)/q_a)_{a\,\varepsilon\,\Sigma,\ i=1\ldots L}$ is called a *position specific score matrix* (PSSM) or a *position weight matrix* (PWM)

Count matrix for PSSM from multiple alignment

```
aaaaaaaAaaAAaaAa
-------|--||--|-
SEGEWQLVLHVWAKVE
SMNRQEISDLCVKSLE
SAQGREIITQCFENPH
TCAQIHLVRALWRQVY
NSYQKSIVRNAWRHMS
SYRDFFTLKNWWKSVD
PKLDIDRVRSVWMDHI
LGDRLSILKSSWEKAN
SDRQRDVLQKTFAPIL
TRRERILLEQSWRKTR
```

Multiple alignment of helix a of Fig. 1

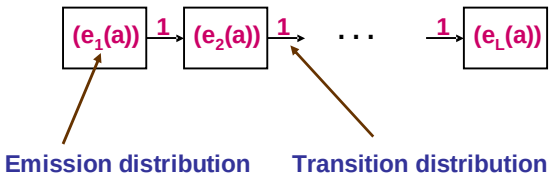
$e_1(S) = 5/10 = 0.5$

$e_1(L) = 1/10 = 0.1$

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

Count matrix (fragment) of helix a of Fig. 1

($e_i(a)$) as a (trivial) HMM

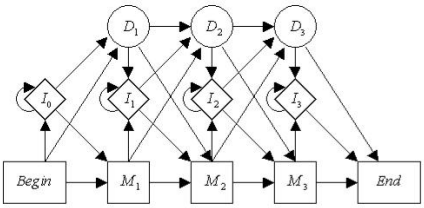


Alignments with gaps and the structure of profile HMMs

```
HBA_HUMAN   ...VGA--HAGEY...
HBB_HUMAN   ...V----NVDEV...
MYG_PHYCA   ...VEA--DVAGH...
GLB3_CHITP  ...VKG------D...
GLB5_PETMA  ...VYS--TYETS...
LGB2_LUPLU  ...FNA--NIPKH...
GLB1_GLYDI  ...IAGADNGAGV...
               ***  *****
```

'Backbone' = columns (*) that correspond to the conserved core of the sequence family to be modeled;

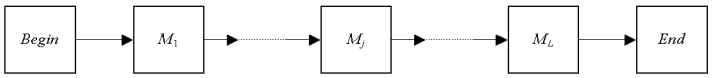
Other columns are needed to represent insertions



Transition structure of a profile HMM

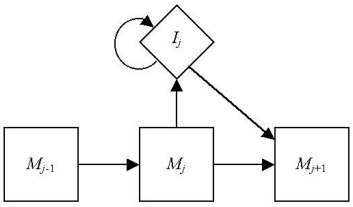
Backbone: match states

- Match states emit the symbols that belong to the 'backbone' of the model



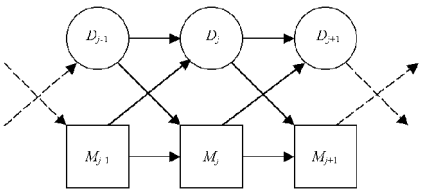
Insert states

- Each Insert state can emit between two match states any number of symbols that do not belong to the backbone model



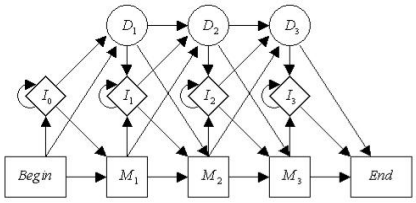
Delete states

- Delete states are needed to present 'jumps' (gaps) that pass some backbone states in an efficient way. This could be done with direct transitions but that would introduce a large number of parameters. Therefore the structure shown below is normally used.
- Delete states are *silent* (do not emit any symbol).



Profile HMM: standard structure

- All HMM algorithms (Viterbi, Forward, Backward, Baum-Welch training etc) can be adapted for the profile HMM



Profile HMM for global alignment

Learning profile HMMs from alignments

- **Input: Multiple alignment λ of some sample sequences from the sequence family to be modeled by the profile HMM**
- **1. Select some columns 1, ..., L of the alignment λ to the backbone; these will correspond to the match states M_1, ..., M_L of the profile HMM**
  - **Take the best conserved columns, with no gaps**
- **2. Estimate probabilities a_{kl}, $e_k(a)$**

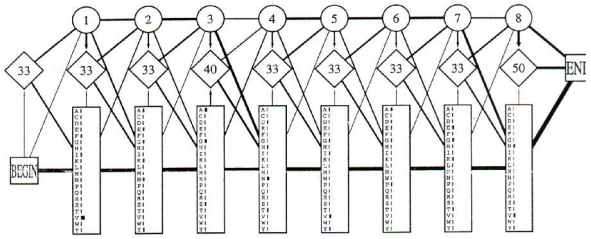
$$a_{kl} = \frac{A_{kl}}{\sum_{q \in Q} A_{kq}} \qquad e_k(b) = \frac{E_k(b)}{\sum_{\sigma \in \Sigma} E_k(\sigma)}$$

where
  - **A_{kl} = (the count of transitions k→l in λ) + 1 (= Laplace rule of pseudocounts)**
  - **$E_k(a)$ = (the count of emissions of a from state k in λ) + 1**

Learning a profile HMM: an example

```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                ***  *****
```

Ten columns from the multiple alignment of seven globin protein sequences. The starred columns are ones that will be treated as 'matches' in the profile HMM.

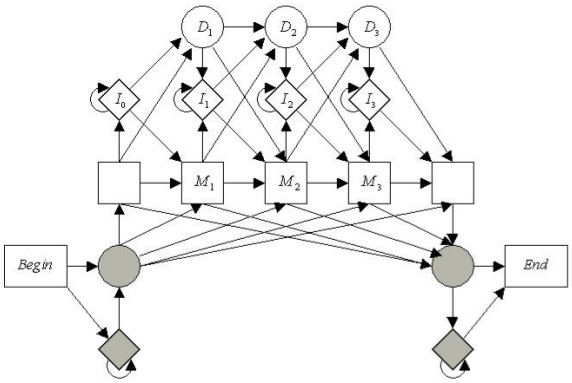


A HMM derived from the alignment using Laplace's rule (add pseudocount 1 to each count). Emission probabilities shown as bars opposite the different amino acids for each match state, transition probabilities indicated by the thickness of the lines. The *I* → *I* transition probabilities are shown as percentages in the insert states.

Aligning Sequences to a Profile HMM

- Alignment of a sequence against a profile HMM: find Viterbi path

Profile HMM for local alignment



What a multiple alignment means?

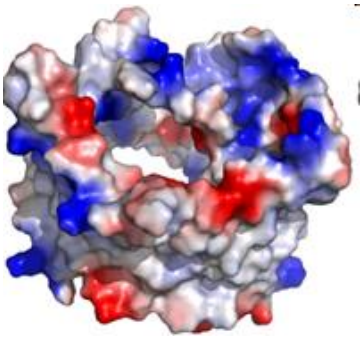
- Biologically correct *multiple sequence alignment* is in general not unique, or at least, there is no precise definition of what is the best alignment
- A correct alignment should align the substructures (for example: alpha helices, beta strands) of different molecules such that the structures that correspond to each other become on top of each other. Such 3D-substructures do not, however, always have clear boundaries in the sequence, or they are not known accurately
- Hence it is difficult to design an automatic alignment method that would produce biologically correct results.
- Alignment algorithms are typically based only on the sequences as such (and possibly on their evolutionary trees) but not on additional information on the locations of the 3D-substructures

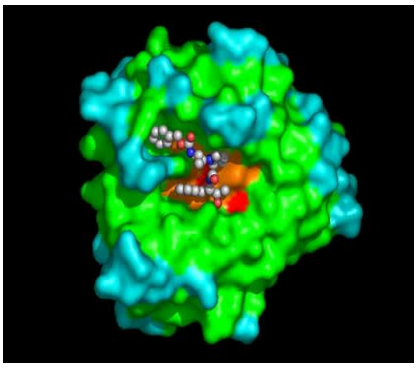
Why multiple alignments?

- **Example**: The Pfam database (pfam.sanger.ac.uk) is a large collection of protein families, each represented by ***multiple sequence alignments*** and ***hidden Markov models*** *(HMMs)* and HMMs visualized as *HMM logos;* *http://pfam.sanger.ac.uk/family/PF00178#tabview=tab0*

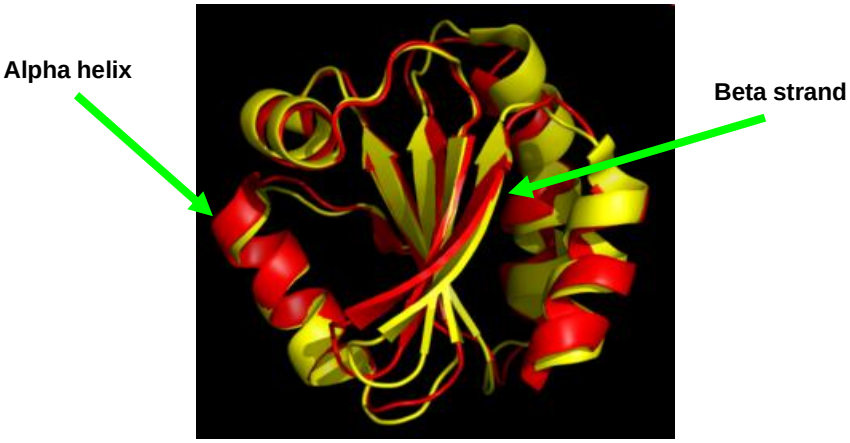
- Goal of multiple alignment: put homologous residues (amino acids, bases) among a set of sequences together in columns
- Homologous = structurally homologous or evolutionary homologous
- Structural = 3D shape
- Evolutionary = conservation in evolution

Protein 3D structure





Structural alignment



Structural alignment of thioredoxins from humans and the fly Drosophila melanogaster. The proteins are shown as ribbons, with the human protein in red, and the fly protein in yellow.

Multiple Sequence Alignment of Globins

Alignment of 9 representative globins and *Sperm whale* (SW) myoglobin. Eight alpha helices are shown as a-h above the alignment. Numbers between brackets indicate the number of amino acids preceding and following the globin domain. [Hoogewijs *et al. BMC Genomics* 2007 **8**:356]

Automatic alignment

- Manual multiple alignment is tedious
- Automatic multiple alignment
  - Biologically 'correct' alignment difficult
  - Important to align the conserved/stucturally similar residues correctly, the areas in between less important; position specific scoring
  - Typical data = the sequences (no annotations such as structural information)
  - Algorithmic challence

Multiple alignment with a known profile HMM

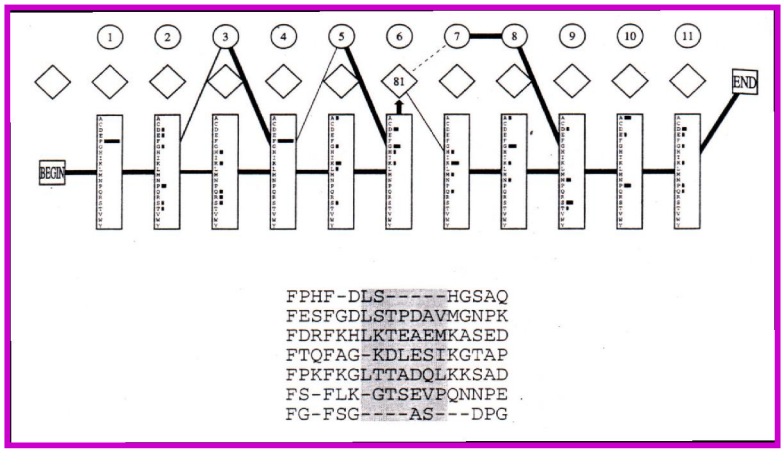
If the profile HMM *M* is known, the following procedure can be applied to generate multiple alignments:

• Align each sequence *S*(*i*) to the profile *M* separately (Viterbi path!)

• Accumulate the obtained alignments to a multiple alignment.

• Insert runs are not aligned, i.e. the choice of how to put the letters in the insert regions is arbitrary (Most profile HMM implementations simply left-justify insert regions, as in the following example).

Example: another profile HMM



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

Alignment generated with the profile HMM

```
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
```

Left: **The alignment of seven sequences generated with the profile HMM of the previous slide. Lower-case letters mean inserts, and the dots are just space-filling characters to make the matches line up correctly.**

Right: **The alignment 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.**

Simultaneous estimation of a profile HMM and multiple alignment from unaligned training sequences

If the profile HMM *M* is not known, one can use the following technique in order to obtain a profile HMM from the given sequences X:

• Choose a length *L* for the profile HMM and initialize the transition and emission probabilities.

• Train the model using the Baum-Welch algorithm and using the sequences X as the training sequences.

• Obtain the multiple alignment of sequences X from the resulting profile HMM, as in the previous case.

HMM Logo: example

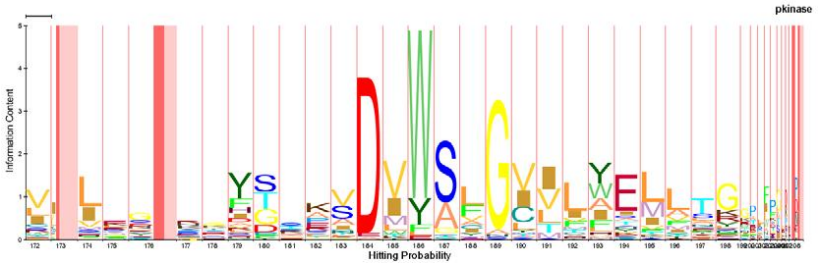


Figure Partial logo (positions 172–209) of the Pfam pkinase model. Positions with narrow match state stacks are likely to be deleted in typical family members. The total width of a red-shaded (dark+light) stack visualizes the expected number of inserted letters.The left dark-shaded part of the stack's width represents the probability that at least one letter is inserted. The difference is illustrated by comparing I_{173} with I_{176}: Both states have approximately the same expected contribution, but the hitting probability of I_{176} is higher. The insertion stack height is zero for all shown examples because the emission probabilities correspond to the background frequencies.
PFAM: http://pfam.sanger.ac.uk/family/PF00178#tabview=tab0

Multiple alignment by multi-dimensional dyn. programming

- Generalization of 2-dimensional dynamic programming to N sequences
- Linear gap score (affine model also possible but tedious to formulate)
- Multiple alignment problem

  Given sequences

  $x^{(1)} = x_1^{(1)}...x^{(1)}_{L(1)}, \quad \ldots, \quad x^{(N)} = x^{(N)}_1...x^{(N)}_{L(N)}$

  find a multiple alignment m for the sequences such that $\sum_i S(m_i)$ is maximum; $S(m_i)$ is the score of the i^{th} column of m.

Algorithm (multi-dimensional Needleman-Wunsch)

- $\alpha_{i(1),i(2),...,i(N)}$ = score of the best alignment of prefixes of length $i_1, i_2, ..., i_N$ of the N sequences

1. $\alpha_{0,...,0} := 0$
2. For $(i_1,...,i_N) := (1,0,...,0), ..., (L(1),L(2),...,L(N))$ do

$$\alpha_{i_1,...,i_N} := \max \begin{cases} \alpha_{i_1-1,...,i_N-1} + S(x^{(1)}_{i_1},...,x^{(N)}_{i_N}) \\ \alpha_{i_1,i_2-1,...,i_N-1} + S(-,x^{(2)}_{i_2},...,x^{(N)}_{i_N}) \\ \alpha_{i_1-1,i_2,i_3-1,...,i_N-1} + S(x^{(1)}_{i_1},-,x^{(3)}_{i_3}...,x^{(N)}_{i_N}) \\ e.t.c. \end{cases}$$

3. $S(m) := \alpha_{L(1),...,L(N)}$

Time and space

- Let all L(i) ≤ L
- Time: $O(2^N L^N)$
- Space: $O(L^N)$
- Too much!
- Optimal multiple alignment (for SP score) is NP-complete (Wang&Jiang 1994)

Divide-and-conquer heuristics

- each sequence is cut in two behind a suitable cut position somewhere close to its midpoint
- therefore the problem of aligning one family of long sequences is divided into the two problems of aligning two families of shorter sequences
- this is re-iterated until the sequences are sufficiently short
- optimal alignment by Carillo-Lipman MSA
- finally, the resulting short alignments are concatenated

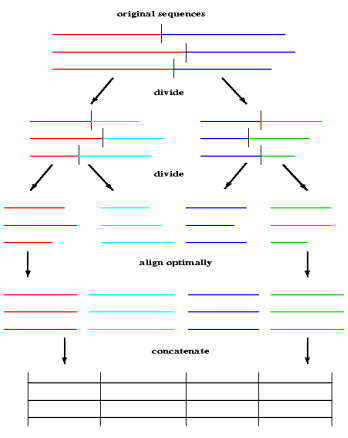


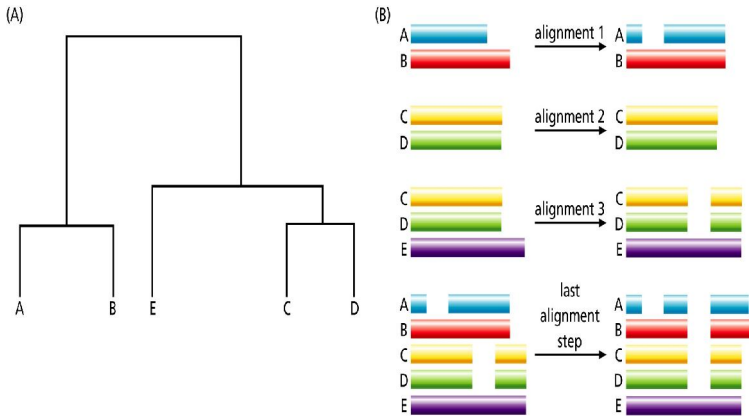
Figure: Divide-and-Conquer Alignment (schematic)

- J.Stoye, *Gene* 211(1998) GC46-CG56;
- Sammeth, Morgenstern & Stoye, *Bioinformatics* 19, suppl 2 (2003) ii89-ii95.

Progressive alignment methods

- These (greedy) methods are the most commonly used approach to multiple sequence alignment. The general idea:
  - Most progressive alignment algorithms build a "**guide tree**", a binary tree whose leaves represent sequences and whose interior nodes represent alignments. (The methods for constructing guide trees can be "quick and dirty" versions of those for phylogenetic trees.)
  - **Main heuristic**: first align the most similar pairs of sequences, using a pairwise alignment method. Then walk up the tree and compute at each interior node the alignment of (alignments of) sequences associated with the direct descendants of that node.
  - The root node will represent a complete multiple alignment of the input sequences.
- Progressive alignment methods use no global scoring function of alignment correctness.

Alignment with a guide tree



Progressive alignment: CLUSTALW

- Construct a distance matrix of all N(N-1)/2 pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity scores to evolutionary distances using the model of Kimura
- Construct a guide tree by using the Neighbour-Joining clustering algorithm [Saitou & Nei]
- Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.
- Apply several additional heuristics to improve the result:
  - sequences are weighted according to the branch length in the tree;
  - BLOSUM80/BLOSUM50;
  - position-specific and dynamically changing gap penalties;
  - dynamically changing guide tree

- T-COFFEE (successor of CLUSTAL): Notredame, Higgins, Heringa (*J. Mol. Biol.* 302 (2000),205-217)

Different (heuristic) alignment algorithms give different results