Searching for family members - (Durbin et al., Ch.5)

• Suppose we have a family of related sequences
  • interested in searching the db for additional members

• Lazy ideas:
  • choose a member
  • try all members

• In either case we are loosing information
  • better: combine information from all members

• The first step is to create a multiple alignment
Multiple alignment of seven globins

Helix

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-DLS------</td>
<td>-GDLSTPDAVMGPKVAHGGKVKLGAFSDGLAHLD-----</td>
<td>-KHLKTEAEMKASEDLKKGVTLTLGAILKK--K--</td>
<td>-AG--KDLKSIKGTPETHANIRVGFFSKS&lt;IGEL--P--</td>
<td>-KGLTTADQLKKSSADVRS&lt;WAERIINAVNDAVASM-</td>
<td>-LK-GTSEVPQNNPHELQAHAGKVFKLVEEAIQLQV&lt;TVGT&lt;VTADTLKLNLSVHVSKG-</td>
<td>-SG--AS---</td>
</tr>
</tbody>
</table>

Consensus

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-RVDPNFLSHELCLVLTLAHLPAEFTRPAVHASLDKFLASVSTVLTSKRY-</td>
<td>-HVDPENFLSLGVCLMVLAHFGEFTRPVPNQAYVQKVAGVANALAHKYH--</td>
<td>-KIPIKYLEFISEAIIIHVLSHRPGDFGADAQGAMNKAELFRKDIAAKYELGYQG</td>
<td>-VTHDQLNFRAGFVSYMKHT--</td>
<td>-QVDPQYFKVLAAVIADTVAG--</td>
<td>-VADAHFPVVKEAILKTKEVGGW&lt;SEELNSAWTIAYDELAIVIKKEMNDAA--</td>
<td>-KHIAQYFEPGLASLSAMEHRIGKMNAAAKD&lt;AAAYADGISGAQLGKS--</td>
</tr>
</tbody>
</table>

Consensus

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-LVSLPDNTNVKAAWVKGA--HAGEYGAELERMFLSPFTTKYTFPHF-</td>
<td>-VHLPDKTAVLWGKV---NVDEVGGEALGRLLVYFWTQRFESF</td>
<td>-VLSEGFWQLLHVWAKVEA--DVAGHGDILIRLFKSHPETLEKFDRF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consensus

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-LVSLPDNTNVKAAWVKGA--HAGEYGAELERMFLSPFTTKYTFPHF-</td>
<td>-VHLPDKTAVLWGKV---NVDEVGGEALGRLLVYFWTQRFESF</td>
<td>-VLSEGFWQLLHVWAKVEA--DVAGHGDILIRLFKSHPETLEKFDRF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consensus

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-LVSLPDNTNVKAAWVKGA--HAGEYGAELERMFLSPFTTKYTFPHF-</td>
<td>-VHLPDKTAVLWGKV---NVDEVGGEALGRLLVYFWTQRFESF</td>
<td>-VLSEGFWQLLHVWAKVEA--DVAGHGDILIRLFKSHPETLEKFDRF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consensus

<table>
<thead>
<tr>
<th>HBA_HUMAN</th>
<th>HBB_HUMAN</th>
<th>MYG_PHYCA</th>
<th>GLB3_CHITP</th>
<th>GLB5_PETMA</th>
<th>LGB2_LUPLU</th>
<th>GLB1_GLYDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-LVSLPDNTNVKAAWVKGA--HAGEYGAELERMFLSPFTTKYTFPHF-</td>
<td>-VHLPDKTAVLWGKV---NVDEVGGEALGRLLVYFWTQRFESF</td>
<td>-VLSEGFWQLLHVWAKVEA--DVAGHGDILIRLFKSHPETLEKFDRF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Profile and Position Specific Scoring Matrix

• In this section we assume the alignment is given
  ● by structure alignment or multiple sequence alignment

• Ignore insertions/deletions for now

• Each position in the alignment has its own “profile” of conservation

• How do we score a sequence aligned to the family?

• Use these conservation profiles to define PSSMs, or Position Specific Scoring Matrices
Gribskov et al.’s PSSMs (87)

- One approach is to average the contributions from the substitution matrix:
  \[ s_i(k) = \sum_j \alpha_{ij} S(k, j) \]
  - \( \alpha_{ij} \) is the frequency of the \( j \)th AA at the \( i \)th position
  - \( S(k, j) \) is the score of substituting AA \( k \) with \( j \)

- If the family contains just one sequence (pairwise alignment) the profile degenerates to one letter, \( x_i \), and
  \[ s_i(k) = S(k, x_i) \]
  - which is exactly the scoring matrix we use for pairwise alignment

- A downside of this approach is that it fails to distinguish between a degenerate position 100 letters “deep” vs. 1 letter deep
HMM’s derived PSSMs (Haussler et al. 93)

- An alternative approach is to think about the positions as states in an HMM each with its own emission profile: \( p(x) = \prod_i e_i(x_i) \)
  - At this point there is nothing hidden about this HMM
- To test for family membership we can evaluate the log-odds ratio

\[
S = \sum_i \log \frac{e_i(x_i)}{q(x_i)}
\]

- the PSSM \( s_i(x) := \log \frac{e_i(x)}{q(x)} \) replaces the substitution matrix
- The emissions probabilities can be quite flexible
  - For example, in the case of a 1-sequence family we can set
    \[
e_i(x) := \frac{p(x,x_i)}{q(x_i)}
    \]
    - where \( p(x, y) \) is the joint probability from BLOSUM
  - and \( s_i(x) = \log \frac{p(x,x_i)}{q(x)q(x_i)} = S(x, x_i) \) as for pairwise alignment
Mind the gap

- How should we handle gaps?
- Gribskov et al. suggested a heuristic that decreased the cost of a gap (insertion or deletion) according to the length of the longest gap, in the multiple alignment, that spanned that column
  - this (again) ignores the popularity of the gap
- Alternatively, we can build a generative model that allows gaps
“Evolution” of profile HMMs

• Profiles without gaps; match states emit according to $e_M(x)$

![Diagram of HMM with Begin, \ldots, M_j, \ldots, End]

• Allowing insertions; for insert states emissions $e_I(x) = q(x)$ typically

![Diagram of HMM with Begin, \ldots, M_j, \ldots, End with I_j insert]

• using llr the score contribution of a $k$ letter insert is

$$\log a_{M_jI_j} + (k - 1) \log a_{I_jI_j} + \log a_{I_jM_j}$$

corresponding to an affine gap penalty (in pairwise alignment)
Evolution of profile HMMs - cont.

- Allowing for deletions

- Too many parameters: recall the silent states

  - the cost of $D_i \rightarrow D_{i+1}$ can vary

- Profile HMMs (Haussler et al. 93):
Deriving profile HMMs from multiple alignment

The first problem in deriving the profile HMM is that of determining the length, or the number of gap states.

Heuristic: a column is a match state if it contains < 50% gaps

- for 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...
```

With the topology of the HMM given the path generating every sequence in the family is determined.

We can use maximum-likelihood with pseudo-counts to estimate the parameters: $a_{kl}$ and $e_k(x)$. 
Example of parameters estimation

Using Laplace’s rule (add a pseudocount of 1 to each count) we have, for example, for the emission probabilities at $M_1$:

$$e_{M_1}(X) = \begin{cases} 
\frac{6}{27} & X = V \\
\frac{2}{27} & X \in \{I,F\} \\
\frac{1}{27} & X = AA \text{ other than } V, I, F
\end{cases}$$

Similarly, using the same pseudocounts, we estimate the transitions out of $M_1$ by:

$$a_{M_1M_2} = \frac{7}{10}, \quad a_{M_1D_2} = \frac{2}{10}, \quad \text{and} \quad a_{M_1I_2} = \frac{1}{10}$$
Searching with profile HMMs

- To determine whether or not a new sequence belongs to the family we need a similarity criterion
  - analogous to the similarity score Needleman-Wunsch optimizes
  - We can ask for the joint probability of the ML path and the data
  - or, for the probability of the data given the model
  - In either case for practical purposes log-odds ratio is preferable

- Reminder: profile HMMs
Viterbi equations (from Durbin et al.)

- Let $V_j^s(i)$ be the log-odds ratio of the best path matching $x_{1:i}$ to the model that ends at state $s_j$ ($s \in \{M, D, I\}$). For $j \geq 1$:

- Initial conditions: $V_0^M(0) = 0$ and $V_0^I = \log \frac{e_{I_0}(x_0)}{q_{x_0}} + \log a_{M_0I_0}$
- An end state needs to be added
- Similar to NW, only scores are position dependent
Forward algorithm (from Durbin et al.)

- For $s \in \{M, D, I\}$ let $F_j^s(i) = \log \frac{P_M(x_{1:i}, S_{\text{last}} = s_j)}{P_R(x_{1:i})}$

$$
\begin{align*}
F_j^M(i) &= \log \frac{e_{Mj}(x_i)}{q_{xi}} + \log \left[a_{Mj-1Mj} \exp \left(F_{j-1}^M(i-1)\right)
+ a_{Ij-1Mj} \exp \left(F_{j-1}^I(i-1)\right) + a_{Dj-1Mj} \exp \left(F_{j-1}^D(i-1)\right)\right]; \\
F_j^I(i) &= \log \frac{e_{Ij}(x_i)}{q_{xi}} + \log \left[a_{Mj-Ij} \exp \left(F_{j}^M(i-1)\right)
+ a_{Ij-Ij} \exp \left(F_{j-1}^I(i-1)\right) + a_{Dj-Ij} \exp \left(F_{j-1}^D(i-1)\right)\right]; \\
F_j^D(i) &= \log \left[a_{Mj-1Dj} \exp \left(F_{j-1}^M(i)\right) + a_{Ij-1Dj} \exp \left(F_{j-1}^I(i)\right)
+ a_{Dj-1Dj} \exp \left(F_{j-1}^D(i)\right)\right].
\end{align*}
$$

- As before $P_R(x) = \prod_i q_{xi}$
- $F_0^M(0) = 0$
- $\log(e^x + e^y) = x + \log(1 + e^{y-x})$ and assuming wlog $y < x$ one can use a tabulated $\log(1 + h)$ for small $h$
Example: searching for globins

- 300 randomly picked globin sequences generated profile HMM
- SWISS-PROT (r.34) which contained ∼ 60,000 proteins was searched
- using the forward algorithm for computing both LL and LLR
  - the null model was generated from the training set
- Note the difference in the variance and normalization problems
• Choosing a cutoff of 0 for the LLR will lead to many false negatives:
  • the training set is not sufficiently diverse

• Can use Z-scores to fix that:
  • fit a smooth “average” curve to each of the non-globins graphs
  • estimate a “local” standard deviation (use a small window)
  • replace each score $s_i$ by $\frac{s_i - \mu(l_i)}{\sigma(l_i)}$

• LLR is a better predictor: without normalizing sequences with a similar composition to globins tend to score higher
Finding the average curve - moving average

- The data is modeled as random fluctuations about a deterministic curve
- The original approach by Krogh et al. (94) used windows of roughly 500 non-globin sequences of similar length
- The scores and lengths in each window were averaged
- The average curve is the piecewise linear curve connecting the averages
- Linear regression was used in the first and last windows
- Standard deviations are computed per window
- Remove outliers, re-estimate average curve and iterate
- This is a slight modification of the moving average method
Finding the average curve - LOWESS and LOESS

- LOWESS and LOESS (Cleveland 79,88) - locally weighted regression and smoothing scatter plot
  - use locally weighted polynomial regression to smooth data
    - or, build the deterministic part of the variation in the data
- At each point (length) $x_0$ of the data consider only the data in $N_{x_0}$, a local neighborhood of fixed size about $x_0$
  - regress data in $N_{x_0}$ on first (LOWESS) or second (LOESS) degree polynomials
  - use weighted regression, with $d := d(x_0) := \max_{x \in N_{x_0}} |x - x_0|

\[
\text{tri-cube: } w(x) = \begin{cases} 
[1 - (\frac{x-x_0}{d})^3]^3 & |x - x_0| < d \\
0 & |x - x_0| \geq d
\end{cases}
\]

- Weighted regression: find $\min_f \sum_i w_i |y_i - f(x_i)|^2$