Profile Analysis

Lecturer: Marina Alexandersson

11 November 2002

1 Introduction

Most functional sequences come in families having diverged from a common ancestor either through duplication within the genome, or through speciation into different organisms.

Sequences within a family usually have the same or a similar function, thus placing a new sequence into a family infers information about its function.

We can use a sequence family to search a database for new members. One way would be pairwise searches for each sequence in the family, but then we might miss distantly related sequences matching features in the multiple alignment but not in the pairwise.

We want to use features common for the family as a whole. Profile analysis is about identifying and using such features.

Example

The protein patter of Cu/Zn Superoxide Dismutase can be written as

[GA] – [IFAT] – H – [LIVG] – H – x(2) – [GP] – [SDG] – x – [STADG]

- Each position in pattern is separated by a hyphen "–".
- x means "any residue".
- [GA] means "G or A".
- { } surrounds forbidden residues (none in example).
- ( ) surrounds repeat counts
- < or > signifies the beginning or end of the protein sequence.

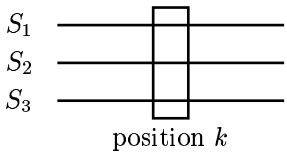
H stands for a copper ligand in this example.

Motif = biologically conserved regions.
Pattern = motif description based on a given syntax.
Profile = motif description based on a weight matrix.

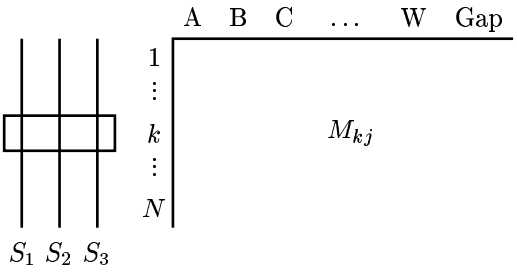
2 What is a profile?

A profile is a position specific scoring matrix (PSSM) that has N rows and 20+ columns. N is the length of the protein motif, and there is a column for each possible amino acid and usually one for gaps.

In a multiple alignment each row is a sequence and each column a position in th4e multiple alignment.



In a profile each row is now a position in the motif or multiple alignment, and each column is the score of an amino acid at each position in the motif.



M_{kj} = score for observing amino acid j at position k in the motif.

Recall that for scoring matrices we used

$$S_{ij} = \ln \frac{p_{ij}}{p_i p_j}.$$

A profile is similar to this

$$M_{kj} = \ln \frac{p_{kj}}{p_j}$$

where i, j are amino acids and k the position in the profile.

3 Average Score Method

The Average score method uses profile scores

$$M_{kj} = \sum_{i=1}^{20} \frac{C_{ki}}{Z} S_{ij}$$

where

C_{ki}	=	# amino acid i at position k
Z	=	total number of aligned sequences
S_{ij}	=	similarity score from PAM or BLOSUM

Example

Assume we have the following multiple alignment

			$k=3$						
S_1	T	H	F	W	K	C_{3F}	=	2	
S_2	S	R	W	Y	R	C_{3W}	=	1	
S_3	S	R	F	Y	R	C_{3M}	=	1	
S_4	S	K	M	W	K	C_{3i}	=	0	for other i.

$$
\begin{aligned}
M_{3F} &= \frac{1}{2}S_{FF} + \frac{1}{4}S_{WF} + \frac{1}{4}S_{MF} \\
M_{3W} &= \frac{1}{2}S_{FW} + \frac{1}{4}S_{WW} + \frac{1}{4}S_{MW} \\
M_{3M} &= \frac{1}{2}S_{FM} + \frac{1}{4}S_{WM} + \frac{1}{4}S_{MM} \\
M_{3i} &= \frac{1}{2}S_{Fi} + \frac{1}{4}S_{Wi} + \frac{1}{4}S_{Mi}.
\end{aligned}
$$

Profile values for two un-observed residues Y and E:

$$
\begin{aligned}
M_{3Y} &= \frac{1}{2}S_{FY} + \frac{1}{4}S_{WY} + \frac{1}{4}S_{MY} \\
M_{3E} &= \frac{1}{2}S_{FE} + \frac{1}{4}S_{WE} + \frac{1}{4}S_{ME}
\end{aligned}
$$

M_{3Y} is much more favorable than M_{3E} even though neither Y nor e has been seen at this position. The reason is that Y is similar to the three types of amino acids F, W and M and gets a higher S-score.

4 Pseudocounts

It makes sense to use as profile as scores

$$M_{kj} = \ln \frac{p_{kj}}{p_j}$$

where

$$
\begin{aligned}
p_{kj} &= \Pr(\text{residue } i \text{ occurs at position } k) \\
p_j &= \Pr(\text{residue } j) \quad \text{(background probability)}
\end{aligned}
$$

As the total number of aligned sequences (Z) grows $p_{kj} \approx \frac{C_{kj}}{Z}$, but for small Z this is a poor estimate for p_{kj}. For residues not yet seen we get $p_{kj} = 0$. The reason to why we haven't counted that amino acid in that position yet might be because we have too few sequences rather than that amino acid is impossible in that position.

One way around this is to add "fake counts", called *pseudocounts*, to avoid zero counts. For instance one can add 1 extra to all counts.

5 Bayesian prediction method

The simplest approach uses as estimate for p_{kj}

$$p_{kj} = \frac{C_{kj} + Bp_j}{Z + B}$$

where B can be seen as a pseudocount taken into proportion to p_j.
Using $B \approx \sqrt{Z}$ has proved to be efficient, but this estimate ignores similarities between observed and unobserved amino acids. Both E and Y in the example above would get the same score, even though Y is much more similar to F, W and M.

6 Data-dependent pseudocounts

Th3 pseudcount Bp_j can be set to depend on observed amino acids at a certain position via a scoring matrix S_{ij}

$$p_{kj} = \frac{C_{kj} + Bp_j \sum_{i=1}^{20} \frac{C_{ki}}{Z}\, e^{\lambda S_{ij}}}{Z + B}$$

where λ is a constant that gets rid of the dependence on scoring system (same as for BLAST).

7 PSI-BLAST (Position Specific Iterated BLAST)

The pseudocount frequencies in PSI-BLAST are set to

$$g_{kj} = p_j \sum_{i=1}^{20} f_{ki}\, e^{\lambda S_{ij}}$$

and

$$p_{kj} = \frac{\alpha f_{kj} + \beta g_{kj}}{\alpha + \beta}$$

where f_{kj} is the observed frequency of amino acid j at position k. α and β are relative weights to observed and pseudocount frequencies.

Example

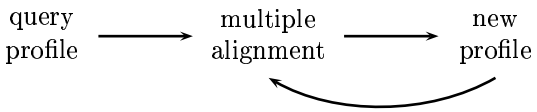
			$k=3$			
S_1	T	H	F	W	K	$f_{3F} = frac12$
S_2	S	R	W	Y	R	$f_{3W} = frac14$
S_3	S	R	F	Y	R	$f_{3M} = frac14$
S_4	S	K	M	W	K	$f_{3i} = 0$ for other i.

$$g_{3F} = p_F\left(\frac{1}{2}\, e^{\lambda S_{FF}} + \frac{1}{4}\, e^{\lambda S_{WF}} + \frac{1}{4}\, e^{\lambda S_{MF}}\right)$$

$$p_{3F} = \frac{\alpha\frac{1}{2} + \beta g_{3F}}{\alpha + \beta}$$

8 BLAST a profile against a database

We begin with a query profile (produced from a multiple alignment), and we search the database for new members getting high scores in this profile. When a new member has been found, it is aligned to the multiple alignment and then a new profile is produced. The new profile is used to search the database again.



The E-value reported by PSI-BLAST is the same as for BLAST, $E = mnKe^{-\lambda S}$, where

m = length of query
n = length of database
S = the dynamic programming score
λ and K depend on residue composition in database and query and on the scoring matrix.

The scoring matrix used is almost always BLOSUM62. $\alpha = N_c - 1$ where N_c is the total "real" counts. β is the total number of pseudocounts (default is 10).