Week 4, class 1: Multiple sequence alignment

•Definition – what is MSA?

• Motivation – why we need MSA?

• How do we measure the quality of MSA?

•How do we compute MSA?

* Material for week 4 is in "Introduction to comp.biology by Sebutal/Meidanis

Multiple sequence alignment

•Up until now we compared pairs of sequences

•Often we are given several sequences that we have to align simultaneously in the best possible way

•Want to know which parts of the sequences are similar and which different

Motivation:

•Multiple sequence similarity suggests a common underlying structure of the protein, a common function, or a common evolutionary source

•Multiple sequence similarity carries more information than pairwise sequence similarity

Definition: multiple sequence alignment- Given sequences $S_1,..S_k$, add spaces to create $S'_1,..,S'_k$, respectively (all of the same length) so that at each position many of the letters are identical (or "similar")

Example: Multiple alignment of: `ACBCDB, CADBD, ACABCD`

```
AC..BCDB
.CADB.D.
ACA.BCD.
```

The common way to present the aligned sequences is to align their positions so characters (or spaces) occupy the same column.

How to measure the quality of the alignment?

Properties of a score function for the multiple alignment.

Once we have the aligned sequences, we sum up the score of all the columns.

To score a column we need a scoring function with #row arguments In the example we need score(-,-,-) for all the possibilities of triplets in the columns. E.g., score(A,-,A), score(C,C,C), etc.

```
AC..BCDB
.CADB.D.
ACA.BCD.
```

We assume that the function is independent of the order of its arguments: score(A,-,A) = score(-,A,A)

We would like to reward similarity of same characters or similar amino acids, and penalize spaces and unrelated residues. A score that satisfies the above is

The **sum-of-pairs** (SP) score.

SP_score(I,-,I,V)= p(I,-)+p(I,I)+p(I,V)+p(-,I)+p(-,V)+p(I,V)

It is defined on columns and is the sum of all pairwise scores of the symbols in the column: p(a,b) is the pairwise score for symbols a and b.

Notice: when many sequences are aligned we might get two or more spaces in one column (see last column in slide 4 example). The common practice is to assign p(-,-)=0

we often draw conclusions about multiple alignment by looking at the pairwise alignments.

Denote the (given) multiple sequence alignment by α

The induced pairwise alignment of sequences S_i and S_j are the sequences S'_i and S'_j, when they are aligned according to α.

The following fact is true only if p(-,-)=0

$$\text{SP-score}(\alpha) = \Sigma\ \text{score}(\alpha_{i,j}) \quad \text{over all } i<j$$

Where $\alpha_{i,j}$ is the pairwise alignment induced by α

```
.CADB.D.
ACA.BCD.
```

Sequences 2 and 3, have this induced alignment, and can be written without the last space (that costs ZERO)

Homework assignment: show that if p(-,-) =0 then SP_score(α) that is computed by summing the scores of all columns (as in the SP_score equation on slide 5), is equal to sum of scores of all induced pairwise alignments.

Induced pairwise alignment is also called projection.

An induced pairwise alignment is not necessarily an optimal pairwise alignment.

AT

A- The 2nd and 3rd sequences are not optimally aligned

-T as it might be cheaper to substitute A and T without

AT the spaces

How to compute MSA?

Dynamic programming?

Given a scoring matrix, we could try to apply dynamic programming, as was done in the pairwise case.

With 2 sequences of length about n letters each we had to fill a table of size n^2, and this was the also the time needed.

With 3 sequences we will need a 3D table of size n^3 and about this runtime.

And with k sequences (each of length n) we need a table of size n^k

The required space is **exponential** in the number of aligned sequences, and the running time is even higher.

Computing the MSA by star alignments.

We need heuristics to compute the MSA. Usually, a heuristic does not guarantee the quality of the resulting alignment , it is faster, and in many cases gives reasonably good answers.

In star-alignment we build multiple alignment based on pairwise alignments between one of the sequences (call it the **center** of the star) and all the other sequences.

Let $S_1,..S_k$ be the k sequences we want to align.

We pick one sequence as the center call it S_c

We will construct the MSA α by all the PSA α_{cj} for $j \neq c$

This means that we have to run k DP (each in time $O(n^2)$)

Total of $O(k* n^2)$ (assuming all sequences are of length n)

MSA by star alignment (cont).

Aggregate PSA by a technique called "once a gap – always a gap". Meaning that if, in optimally aligning S_c with S_1, there should be a gap in Sc (or more), then after the next PSA, aligning S_c with S_2, we align the three sequences adding as few spaces as possible so that all the alignments agree.

Example	Aligning $S_c = S_1$ with S_2-S_5	The MSA
S1=ATTGCCATT		
S2=ATGGCCATT	ATTGCCATT ATGGCCATT	ATTGCCATT - - ATGGCCATT - - ATC - CAATTTT ATCTTC – TT - - ACTGACC - - - -
S3=ATCCAATTTT	ATTGCCATT - - ATC –CAATTTT	
S4=ATCTTCTT	ATTGCCATT ATCTTC -TT	
S5=ACTGACC	ATTGCCATT ACTGACC - -	

MSA by star alignment (cont).
How to pick the center sequence?

1. Maximize sum of pairwise scores
A simple way is to compute all pairwise alignments and then pick the center as the sequence that maximizes the similarity score of itself with all the other sequences.

Maximize Σ sim_score(S_i, S_c) over all $i \neq c$

This takes $O(k^2)$ PSA computations (each involving DP).

2. Find best MSA score
Another way to pick a center, is again, try all sequences as centers and pick one that gives best MSA score.

Guaranteeing the quality of star alignment with SP_scores

We required for SP_score the properties

p(-,-) = 0

p(x,y)= p (y,x) - symmetry

We define the **triangle inequality**

p(x,z) ≤ p (x,y) + p (y,z)

Claim

If our score function, p, obeys the triangle inequality then the SP_score we achieve by using the star MSA is ≤ 2*(k-1)/k *OPT

Where OPT is the cost of an optimal solution for the MSA problem