18.417 Introduction to Computational Molecular Biology

Lecture 5: September 23, 2004 — Scribe: Tony Scelfo
Lecturer: Ross Lippert — Editor: Athicha Muthitacharoen

Local/Multi Alignments

Introduction

Last Time

- Global Alignment

This Time

- Local Alignment Method of aligning two sequences that share a highly common subregion. This is often an important technique becuase it will detect common subregions in two sequences that have poor global alignment scores. Common subregions often come up in cases where two similar genes are found in two different species where the overal DNA sequence is different.
- Affine Gap Penalty Often when local alignments are being done, it is desireable to assign different penalities for gaps than for misalignments in an alignment scoring function. The reason for this is that it is often better to have a large gap than it is to have many misalignments. Large gaps will come up when comparing two species because large functional regions are likely to not change.
- Multialignment Multialignment concerns different techniques to align multiple sequences. We will look at a method to simultaneously score multiple alignments as well as a method to progressive align multiple sequences.

Gene File Formats

In order to obtain DNA sequences, certain files formats are used as standards.

- fasta - most common sequence file format

```
>define
...... (ascii)
...... (ascii)
>define
...... (ascii)
...... (ascii)
```

- asn - another format that isn't as popular

Homeobox Genes

Homeobox Genes are good to study when looking at local alignments. The reason for this is that the region that codes for the Homeobox Gene has been identified in many species and biologists can look for the Homeobox Gene in new species or test the performance of local alignment algorithms.

- Why are two sets of genes similar at some point and different at the rest? Genes are similar at the regions that code for the functional parts of genes. It is common for the sections of a sequence between functional regions to evolve rapidly because a change in such a section does not result in a functional change in the gene.
- In homeobox genes, the homeodomain is highly conserved.
- Global alignment can miss localized correlation if most of the gene is uncorrelated.
- Homeobox genes control what cells will turn into what (bodyplan).
- Correlated region corresponds to the region of the gene that binds to DNA for regulation. Over time, regions don't change much because they are so important.

Local Alignment

For local alignment, we just want to align to a substring, not corner to corner on the schematic grid. Unlike global alignment where we try to find the longest path among

paths between verticies $(0, 0)$ and (n, m), in local alignment we try to find the longest path between arbitrary verticies (i, j) and (i', j') in the edit graph.

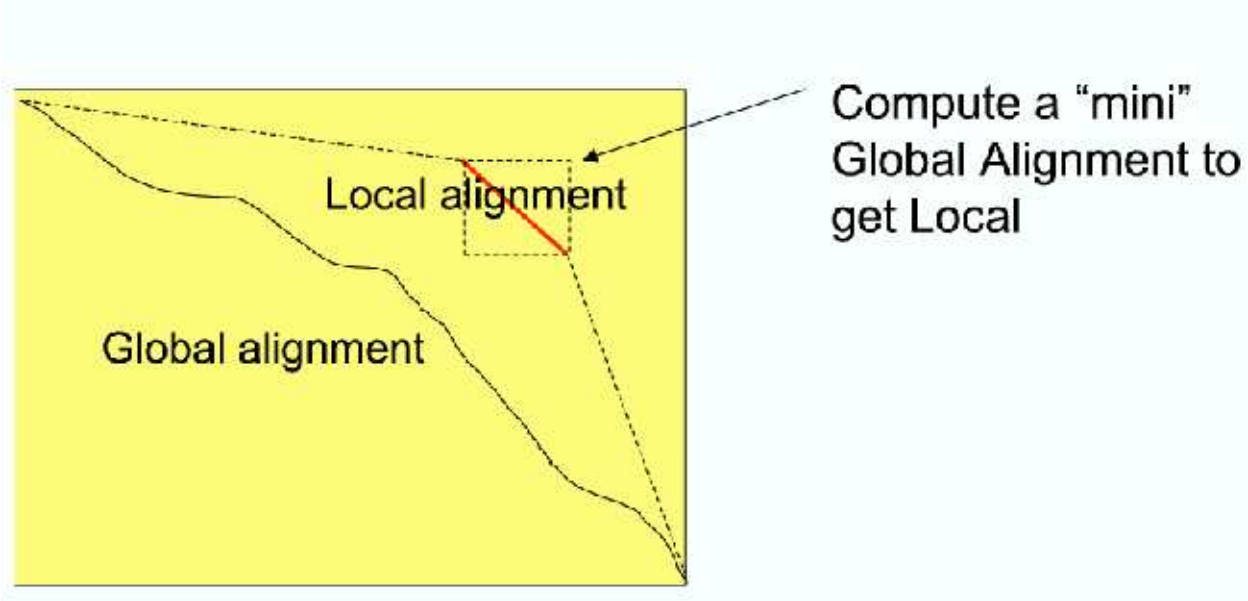

Figure 5.1: Figure showing Local Alignment region

Problem Statement

Input: S_1, S_2, δ
Output: $\max_{i' \leq i, j' \leq j}$ GlobalScore$(S_1[i' \cdots i], S_2[j' \cdots j])$
Time: $O(n^4 \cdot n^2) = O(n^6)$

cell(i,j) = GlobalAlignmentScore$(S_1[1 \cdots i], S_2[1 \cdots j])$

- reduces local alignment to $O(n^4)$.

By using the same grid but different recurrence, can reduce local alignment to $O(n^2)$ (Smith-Waterman).

$$score_{i,j} = max \begin{cases} score_{i-1,j-1} + \delta(S_1[i], S_2[j]) \\ score_{i,j-1} + \delta(-, S_2[j]) \\ score_{i-1,j} + \delta(S_1[i], -) \end{cases}$$

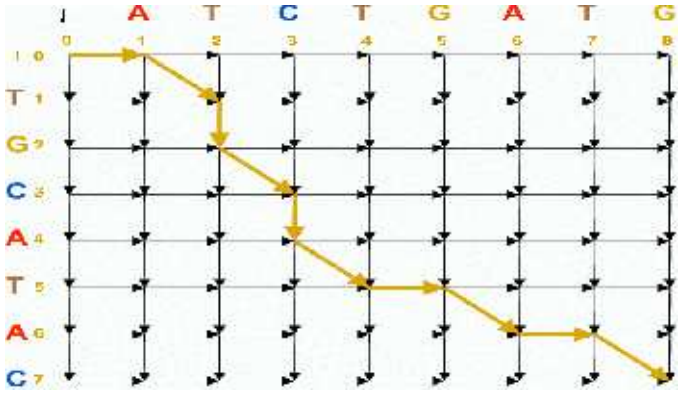

Figure 5.2: LCS Edit Graph for Global Alignment

Affine gap penalties, a common modification

Penalty caused by the energy needed to cause the initial bend (corners). Score alignments = $\lambda \cdot matches - \mu \cdot mismatches - \sigma \cdot gap_c haracters - \rho \cdot total\_continuous\_gaps$

There are now three nodes for each node the schematic grid. The three nodes represent the scores for each point on the edit graph. Gaps allow movements to be made in the edit graph in the horizontal, diagonal and vertical directions as illustrated in the figure.

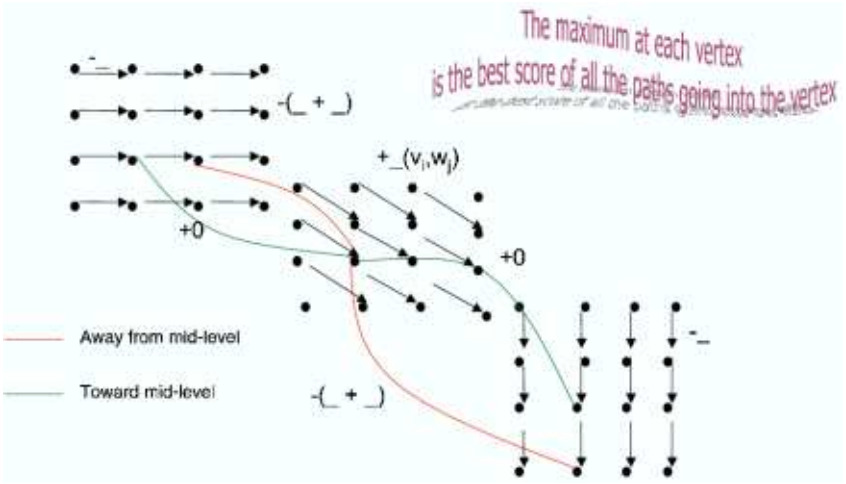

Figure 5.3: Horizontal, Diagonal and Vertical directions

$$H_{i,j} = max \begin{cases} H_{i,j-1} - \sigma \\ D_{i,j-1} - \rho - \delta \end{cases}$$
$$D_{i,j} = max \begin{cases} V_{i,j}, H_{i,j} \\ D_{i-1,j-1} + \delta(S_1(i), S_2(j)) \end{cases}$$
$$V_{i,j} = max \begin{cases} V_{i-1,j} - \delta \\ D_{i-1,j} - \rho - \delta \end{cases}$$

Multiple Alignment

$S_1, S_2, S_3, \cdots, S_d$ are the sequences to be aligned.

$\delta(S_1, S_2, \cdots)$ d-way score measures the distance for all possible pairwise alignments as shown in the example below.

$O(n^d)$ because all possible pairs need to be evaluated.

Three sequence example

d-dimensional $S_{i,j,k}$ = best multi-score $(S_1(1 \cdots i), S_2(1 \cdots j), S_3(1 \cdots k))$

$$S_{i,j,k} = \max \begin{cases} S_{i-1,j-1,k-1} + \delta(S_1(i), S_2(j), S_3(k)) \\ S_{i-1,j-1,k} + \delta(S_1(i), S_2(j), -) \\ \cdots \\ \cdots \text{ 8 ways total} \\ \cdots \end{cases}$$

Performing the alignment

We want to find the value of delta to determine the optimal multi-alignment.

Using information theory $\rightarrow \delta(x, y, z) = \sum_{\{i \in a,t,g,c,-\}} -p_i * log\frac{1}{p_i}$

$\delta(A, A, A, A) = 0$
$\delta(A, T, G, C) = 2$

Sum of pairs: $\delta(x, y)$
$\delta(a_i ... a_l) = \sum_{i<j} \delta(a_i, a_j)$

When aligning k sequences, the running time is $O(2^k n^d)$

Progressive alignment

- e.g. clustal (based on aligning to an alignment)
- does not use ($\frac{\delta}{2}$) pairwise alignments, instead aligns first pair and then aligns next sequence to the existing alignment and then continues until all sequences have been aligned.

Common implementations of progressive alignment algorithms are called Clustal W and Clustal X. Clustal W is the most popular multiple alignment tool today. There are several heuristics that are used to improve accuracy: sequences are weighted by relatedness, scoring matrix can be chosen "on the fly", position-specific gap penalties are used. To perform a progressive alignemnt, the most common sequences are first aligned to each other and then subsequent sequences are aligned to the first two and gaps are inserted appropriately.

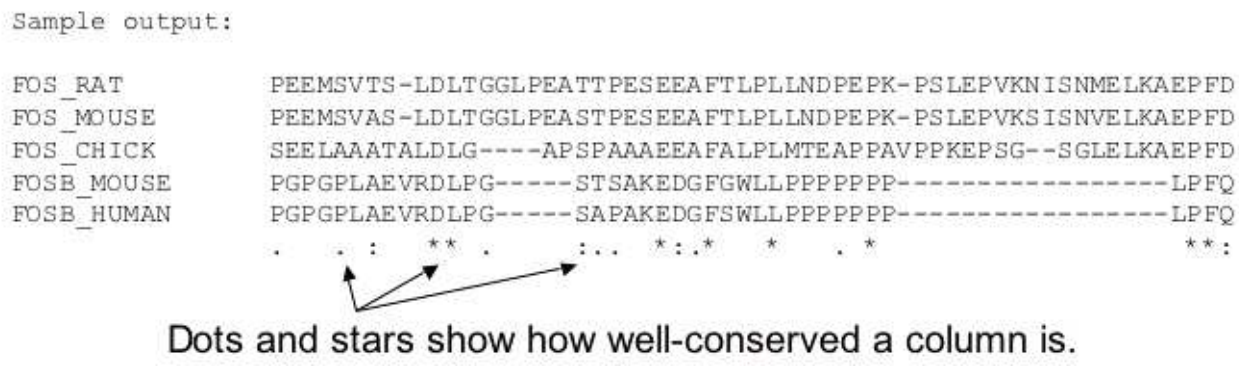

Figure 5.4: Example of Progressive Alignment

What goes wrong with progressive alignment?

- sometimes a bad initial alignment forces bad decisions for the rest of the alignments
- depending on the order that sequences are aligned, the overall alignment will not always be the same