P. Cloé

Gapped motifs in DNA
Rocke-Tompa algorithm for finding
position of the motif removed.

the motif, and using this scores the genome for a new start
computes a probability of nucleotide occurrences at each site of
where one begins with m nonoverlapping motifs, removes one,

Idea similar to Gibbs sampler for multiple sequence alignment,

allowed within non-coding sections of genome.

allowed (within roughly m) since insertions, deletions
mots of size roughly m. Goal is to find a repeat of in gapped

Fix m and window size w. Goal is to find a repeat of in gapped

DNA sequences by Rocke, Tompa in Recomb'98.

Presentation of “An algorithm for finding novel gapped motifs in

Gibbs sampler variant
distribution at site \( j \) with the background distribution.

where \( q \) is the background frequency of residue \( j \). In other

\[
\left( \frac{a/b}{f_{i,j}} \right) \log_2 \sum_{j} f_{i,j}^a = (f, s)_o
\]

by current candidate window, divided by \( m \). Suppose that \( s \) is
alignment of \( m - 1 \) motifs (one has been removed) augmented
occurrence of residue \( j \) at site \( i \) of multiple sequence

Define \( \theta_i \) to be the ratio of the number of
define new scoring function for new start position of motif in
alignment, inductively keep a multiple sequence alignment, and

In place of requiring constant size \( m \) as in multiple sequence
Length (real minus coding regions) 

Naive implementation takes time $O_{\mathcal{C}^m}$, where $\mathcal{C}$ is genuine.

$0 > \langle (\ell, s) \tau \rangle$

Warning. One probably wants

$0 = \langle (\ell, s) \tau \rangle$

so that

$\langle (\ell, s) \sigma \rangle - (\ell, s) \sigma = (\ell, s) \tau$

Recall that relative entropy is non-negative. Authors use alignment (between sequence and a profile) defined by similarity scoring function $T(s')$, for pairwise sequence similarity scoring function $T(s)$. 

$\mathcal{F}$
• Instead, perform modification of pairwise sequence alignment of window of size \( w \) with entire genome \( G \), where first row is initialized to 0 (end-space free alignment). Then score in row \( w \) and column \( i \) is the score of an optimal alignment between the window and a suffix of the \( i \)-letter prefix of \( G \) (i.e. a sequence ending at position \( i \)).

• Time complexity of revised alignment algorithm is \( O(wG) \). Don’t keep entire \( w \times G \) table in memory, but only last part. For traceback, keep roughly \( w \) columns of \( w \times G \) table as progress, trying to find the largest scoring region.

• Evaluation function for \( m \times w \) alignment \( A \) is sum over all columns \( j \) of \( A \) of the score of \( j \). When evaluation function no longer improves, quit.
multiple sequence alignment, for the next iteration, use probabilities \( p \) to determine a new sequence to add to the sequence \( A \).

Alignment of window ending at position \( i \) with remaining alignment of window ending at position \( i \) with remaining alignment \( A \), determining for each site \( j \) of genome \( C \), an optimal alignment \( \alpha \), performing \( O(n) \) dynamic programming pass (end-space free alignment \( \alpha \) of \( m - 1 \) sequences.

This leaves randomly proportioned to evaluation function, or uniformly randomly of maximize evaluation function, or uniformly randomly choosing to choose sequence to remove (by deterministically choosing to \( w \), in genome \( C \).

Alignment of \( m \) nonoverlapping sequences, each of roughly size 1. At beginning of each iteration, have a multiple sequence summary of algorithm
(Deterministically by choosing sequence to maximize $p_i$; else proportionally to probability $p_i$.)