MGALIGN METHODOLOGY

We first define the DNA alphabet $A = \{a, t, g, c\}$ and a gap alphabet $G = \{-\}$. A pairwise alignment is thus made up of two sequences, an mRNA/EST sequence $M = m_1 G^x m_2 G^x \ldots m_n$, $m_i \in A$, $x \in \{0,1,2\ldots\}$ and a genomic sequence $S = s_1 G^x s_2 G^x \ldots s_m$, $s_i \in A$, $x \in \{0,1,2\ldots\}$. To reduce the amount of notional complexity, we represent a subsequence of M as $M(i,j)$ which is equivalent to a subsequence $m_i G^x m_{i+1} G^x \ldots m_j$, $m_i \in A$, $i \geq 1$, $j \leq |M|$, $x \in \{0,1,2\ldots\}$ and similarly, a subsequence of S as $S(i,j)$.

The input to the program comprises an mRNA/EST sequence $M = m_1 G^x m_2 G^x \ldots m_n$, $m \in A$, $x = 0$ and a genomic sequence $S = s_1 G^x s_2 G^x \ldots s_m$, $s_i \in A$, $x = 0$. The output is a presentation of a single alignment consisting of two sequences $M = m_1 G^x m_2 G^x \ldots m_n$, $m \in A$, $x = \{0,1,2\ldots\}$ and a genomic sequence $S' = s_1 G^x s_2 G^x \ldots s_m$, $s_i \in A$, $x = \{0,1,2\ldots\}$ where $|M'| = |G'|$.

First search step

This step determines the location of the 5' and 3' end of the mRNA/EST sequence on the genomic sequence. Firstly, a subsequence $M_{f1}(i,j)$ of M is chosen such that $i = 1$, $j \leq |M|$ and $|M_{f1}| = Z$ (default value is Z=20). This has the effect of taking the 5' most subsequence from M. M_{f1} is then used to locate for subsequence $S_{f1}(o,p)$ in S where $\forall\ m_i = s_o$ and $|S_{f1}| = Z$ to form an alignment of length Z without any gaps. M_{f1} and S_{f1} are then extended using dynamic programming to form a possibly gapped alignment consisting of $M_{f2}(q,r)$ and $S_{f2}(s,t)$ where $|M_{f2}| = |S_{f2}|$. If $q < i$ or $r > j$ is the extension is successful, this alignment $(M_{f2}S_{f2})$ considered further, else it is dropped.

Since there could be multiple $(M_{f2}S_{f2})$, a set $F = \{(M_{f2}S_{f2})_1(M_{f2}S_{f2})_2\ldots\}$ is possible, only in the event that $|F| \leq 10$ and $|F| > 0$, is this set considered further. $|F|$ is limited to 10 to reduce the amount of searching in the second step. Should $F = 0$, then i is incremented by 2, in effect taking a 5' subsequence from M that is 2bp downstream of the previous one and the entire process repeated till $|F| \leq 10$ and $|F| > 0$.

The same procedure is repeated using s subsequence $M_{r1}(i,j)$ except for the fact that this subsequence is the 3' most subsequence of M. This results in a set R, which is similar to F.

$Y = \{((M_{f2}S_{f2})_1(M_{r2}S_{r2})_1),((M_{f2}S_{f2})_1(M_{r2}S_{r2})_2\ldots\}$ is a set consisting of $F \times R$. We term an alignment window w as a set of two alignments $((M_{f2}S_{f2})(M_{r2}S_{r2}))$ where $x - t \leq 1{,}000{,}000$. This is done to minimize the number of false positive alignment pairs. Most genes are no longer than 1 Mbp: by limiting the genomic distance of the alignment pair, the number of false positives is greatly reduced. This can be changed by the user via the "-sws" flag. W is defined as a set of alignment windows w thereby $W \subseteq Y$, $\forall\ w \in Y$, $x - t \leq 1{,}000{,}000$. Each w is then used for the second search step.

The use of a set of alignment windows also serves to handle duplicated and pseudogenes in a manner similar to Spidey. Running the program with the "-allalign" flag provides the results from all the alignment windows.

Second search step

Each alignment window identified in the first search step defines the boundaries for the second search step, which locates the local alignments that map the rest of the mRNA/EST sequence. The search space for the second search step is thus limited to $S_{search}(t+1,x-1)$. To aid the search for the local alignments, a hash table index is created using genomic subsequences $S_{hash}(i,j)$, $i > t$, $j < x$, $|S_{hash}| = 10$, of the same order as the wordsize in BLAST (Z=11) and sim4 (Z=12). The mRNA/EST sequence is then broken into non-overlapping subsequences $M_{hash}(i,j)$, $i > r$, $j < u$, $|M_{hash}| = 10$ and its genomic location determined using the hash table index created. These subsequences are then combined into longer alignment segments if they are found to be adjacent to each other on both mRNA/EST and genomic sequences. Then, non-adjacent subsequences on the same diagonal are joined if the gap between them can be filled. Only subsequences that are paired with at least one other subsequence are then marked for extension using dynamic programming in the next step of selection and arrangement.

Local alignment selection and arrangement

The arrangement and selection of a subset of the local alignment is done using dynamic programming such that the order of the local alignments is the same as that of the mRNA/EST sequence.

Missing sequence search

Should there be any unmatched region between any two adjacent selected local alignments, ($M_1(i,j)S_1(o,p)$) and ($M_2(q,r)S_2(t,u)$), a search is carried out using with 4 bp subsequences $M_{internal}(x,y)$, $x > j$, $y < q$, $|M_{internal}| = 4$ of M within $S_{internal}(p+1,t-1)$. $S_{internal}$ is basically the genomic sequence bounded by the two flanking local alignments. Once a match is found, it is then extended using dynamic programming to cover the entire length of the unmatched segment.

If the unmatched segments are located at the ends of the mRNA/EST sequence i.e. when $i > 1$ in the 5' most local alignment ($M_{5'}(i,j)S_{5'}(o.p)$) or $r < |M|$ in the 3' most local alignment ($M_{3'}(q,r)S_{3'}(t,u)$), a 4-bp long subsequence $M_{flank}(v,w)$, $w < i$ or $v > r$, $|M_{flank}| = 4$ of the M is used to search is carried out till the ends of M. To reduce the number of hits returned due to the short subsequence used, only a 100,000bp subsequence $S_{flank}(x,y)$, $y < o$ or $x > u$, $|S_{flank}| = 100{,}000$ of S is searched, as most introns are shorter than this length. Once again, dynamic programming is used to extend the located match to cover the entire length of the unmatched sequence. If more than one hit is reported from the flanking region, the program selects the nearest hit with a canonical splice site motif. As with the window size, this parameter can be changed using the "-fsws" flag.

Splice site selection

At this point, some of the local alignments need to be trimmed for overlaps. Most local alignments determined at this point can be considered as exons, with splice site motifs contained within the introns that separate these. Thus, overlapping local alignments need to be trimmed such that a canonical splice site motif exists between them, in a manner akin to other programs of this genre. When a non-canonical splice site motif is encountered, the local alignments are simply trimmed to eliminate any overlap.

Alignment window selection and reporting of results

Each alignment window selected in the first search step leads to a final independently determined alignment. Thus at this point, the program needs to select a single alignment window with the largest number of matches. This is the sole alignment that is reported as the default output from MGAlign. Running MGAlign with the “-allalign” flag outputs the alignments contained in all the alignment windows. The alignment is reported in two formats: as a summary listing of the start and end positions of the exons, for easy visual inspection and in a comma delimited format where each output line represents an exon, suitable as input to other analysis/visualization programs.