Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm
Franz Wolfertstetter, Kornelie Frech, Günter Herrmann and Thomas Werner

Abstract
We present an algorithm to identify potential functional elements like protein binding sites in DNA sequences, solely from nucleotide sequence data. Prerequisites are a set of at least seven not closely related sequences with a common biological function which is correlated to one or more unknown sequence elements present in most but not necessarily all of the sequences. The algorithm is based on a search for n-tuples which occur at least in a minimum percentage of the sequences with no or one mismatch, which may be at any position of the tuple. In contrast to functional tuples, random tuples show no preferred pattern of mismatch locations within the tuple nor is the conservation extended beyond the tuple. Both features of functional tuples are used to eliminate random tuples. Selection is carried out by maximization of the information content first for the n-tuple, then for a region containing the tuple and finally for the complete binding site. Further matches are found in an additional selection step, using the ConsInd method previously described. The algorithm is capable of identifying and delimiting elements (e.g. protein binding sites) represented by single short cores (e.g. TATA box) in sets of unaligned sequences of about 500 nucleotides using no information other than the nucleotide sequences. Furthermore, we show its ability to identify multiple elements in a set of complete LTR sequences (more than 600 nucleotides per sequence).

Introduction
Discovery of new functional elements usually necessitates elaborate scanning or deletion mapping experiments not suitable for systematic analysis of large amounts of sequence data. The various genome sequencing projects rapidly produce new sequence data at a rate prohibiting any kind of systematic experimental analysis. However, biological function is defined by the DNA sequence, either directly in the nucleotide sequence (consensus) or by other features like structural elements. Hence, it should be possible to identify and describe a sequence motif responsible for a biological function solely from nucleotide sequence analysis thus reducing experimental efforts to the analysis of few preselected candidates.

Several published search algorithms (Alexandrov and Mironov, 1990; Goodrich et al., 1990; Gartmann and Grob, 1991; Frech et al., 1993) require a priori knowledge about a motif (e.g. approximate position in the sequences and/or IUPAC code of the site) and construct descriptions of varying complexity for these motifs and their flanking regions. However, these programs are not suitable to detect new unknown motifs.

Several methods were published that are capable of identifying statistically significant unknown patterns. However, they cannot select for biologically meaningful tuples without approximately prealigned sequences as input (Galas and Waterman, 1985; Mengeritsky and Smith, 1987) or do not allow mismatches within the pattern (Pesole et al., 1992; Liuni et al., 1993). Other methods based on searching for the highest conserved binding matrix (Stormo and Hartzell, 1989; Hertz et al., 1990) or on Expectation Maximization (Lawrence and Reilly, 1990; Cardon and Stormo, 1992) were shown to identify long sites conserved to a lesser extent (22-nucleotide CRP binding site in Stormo and Hartzell, 1989, and Lawrence and Reilly, 1990, 20 bp LexA binding site in Hertz et al., 1990, E.coli promoters including a variable spacer in Cardon and Stormo, 1992). Though this is not necessarily an upper limit of these algorithms, they have only been shown to work for relatively short sequences (100–200 nucleotides).

Our algorithm (referred to as CoreSearch) is designed to identify consensus regions around short and therefore somewhat higher conserved motifs (tuples) in considerably longer sequences. It is also capable of delimiting the biologically important region of each motif and to recognize additional less conserved elements which are only moderately distance correlated. In addition, it is also able to find multiple uncorrelated motifs in successive runs within the same set of sequences.

We tested our method with seven different sets of
sequences with an average length of about 400 nucleotides. Without any additional information using a single set of parameters CoreSearch identified the correct motifs and their correct extent in all of the 7 sequence sets. We also analysed a set of 17 retroviral long terminal repeat (LTR) sequences with an average length of about 600 nucleotides to show its ability to find more than one motif. CoreSearch found most of the known functional elements in the LTR sequences. CoreSearch also is capable of correctly analysing sets of sequences that contain several highly related sequences.

System and methods

The program has been developed on a DEC Alpha running OSF/1 3.0. The program is completely menu-driven and was designed for ease of use. CoreSearch requires sequence files containing the sequences to be analysed in IG-Suite format (Intelligenetics). Additional input is possible but not required provided the standard parameters are specified. CoreSearch will carry out the complete analysis automatically and produce two different forms of output if a consensus description was built. The first output includes a list of all tuples found and details of the results for each tuple and the other output files are in the format of ConsInd output (Frech et al., 1993) including a binary file that can be immediately used to scan sequences for the consensus with the program ConsInspector.

In order to allow extension of our library of consensus descriptions and to make CoreSearch capabilities available for researchers with no access to DEC Alpha computers we will make the program accessible for the public. However, CoreSearch will be available upon special request as executable file for DEC Alpha (OSF/1). Interested researchers should send the program request or their sequences as ASCII files via e-mail to frech@gsf.de. We will carry out the CoreSearch analysis and provide all result files together with a copy of the program ConsInspector including our complete consensus library ready for use either on a UNIX or VMS system or on a DOS PC (486 required as minimum).

Algorithm

Basic search algorithm

CoreSearch analyses a set of nucleic acid sequences for common elements. It is based on a search for n-tuples which occur at least in a user-defined percentage of the given sequences (n is a parameter). The tuple may contain a single mismatch at a random position. n-tuples identified in this manner will be referred to as ‘basic tuples’. Individual tuples within a basic tuple will be referred to as members of their basic tuple. For example, the tuples TaAGTCA, TGtGTCA, TGAGTCA, and gGAGTCA are all members of the basic tuple TGAGTCA (mismatched nucleotides shown in lower case).

Estimation of the length of the search tuples

The probability for the random occurrence of a basic tuple in the sequences is used for estimation of the search tuple length. The random occurrence probability of any n-tuple with none or a single mismatch in nucleic acids composed of four different nucleotides is:

\[
P_{\text{random}}(n) = \frac{1 + (4 - 1) \times n}{4^n}
\]  

(1)

This simple method assuming equiprobable nucleotides is sufficient for our purposes, because it is only used for estimation of a practicable tuple length and has no influence on the results. According to this formula 0.46% of a 6-tuple and 0.13% by any 7-tuple. The average number of random occurrences (OCCrandom) in a sequence of length l can be calculated by

\[
\text{OCC}_{\text{random}}(n, l) = P_{\text{random}}(n) \times (l - n + 1)
\]  

(2)

n = tuple length

l = length of sequence

In order to avoid results that are obscured by too many random matches OCCrandom must not be close to 1.0. This leads to an upper limit for the sequence length l. For n = 6, l = 150 proved to be practical and for n = 7 the sequence length may be about 500 bases, which results in OCCrandom between 0.6 and 0.7. An OCCrandom above 0.7 is likely to either cause very extensive and time consuming selection procedures or to prevent successful analysis of the data at all. Thus we chose OCCrandom ≤ 0.7 as an empirical limit for the correlation of tuple and sequence length.

For longer sequences a higher n is recommended. However tuples longer than 9 or 10 are not expected to be conserved enough for application of this method nor are they commonly found in DNA elements (Pesole et al., 1992).

We chose sequences containing AP-1 binding sites to test our algorithm, because AP-1 sites represent 7-tuples with some internal variations (Angel et al., 1988).

Example

Analysis of 14 sequences (preselected for presence of one or two AP-1 binding sites)

AP-1 binding sites in the set of 14 sequences:

BPV-1(X02346): TGtGTCA
Identification of functional elements in unaligned nucleic acid sequences

HPV-6(X00203): TGAcTCA
HPV-11(M14119): TGAcTaA
HPV-8(M12737): TGAcTCA
HPV-16(K02718): TGAAaTCA, TtAGTCA
RBFGG(K02708): TGAcTCA
PCGR(K02989): TGAGTCA
HPV-18(X05015): TtAGTCA
HUMCN2A(M16567): TGAGTCA
HPV-31(J04353): TGAGTCA
HUMMET2AA(M15244): TGAcTCA
HPV-33(M12732): TGAcTCA, TGAGTCA
YSCHIS3(X03245): TGAcTcT
SV40XX(V01380): TGATGCA

The 7-tuple TGAGTCA matches in 12 of the 14 sequences with no or with a single mismatch (shown in lower case) to the known AP-1 binding site. Thus TGAGTCA will be identified as a basic tuple (provided the user defined threshold for the number of sequences in which a basic tuple must occur is less than or equal to 12/14, here we assume that the tuple does not occur at other positions). There are only two AP-1 binding sites with two mismatches (marked by an arrow).

Selection of basic tuples

In most cases the initial search algorithm defines more than one basic tuple. In the example (sequences known to contain AP-1 binding sites) 10 basic tuples (ATGTGTTT, CTTGGTTT, GTTGGCA, TGAGTCA, TTATGCA, TTTGCAA, TTTGCA, TTTGCA, TTTTGGC) are defined which occur in more than 90% of the sequences with none or a single mismatch (n = 7, sequence length ~400 bases). Furthermore, in most sequences basic tuples occur more than once with different mismatches. For example the basic tuple TGAGTCA is found as TtAGTCA and TGtGTCA in sequence BPV-1 and as TGAGTCA and gGAGTCA in sequence HUMCN2A (All tuple members of the basic tuple TGAGTCA are listed in Figure 2). However, only a subset of these matches represent true AP-1 binding sites.

Three consecutive selection steps are employed in order to reduce the number of tuple candidates as soon as possible in the order of increasing demand of computer resources. Selection is always based on maximization of the information content (expressed as consensus index, C, calculated from formula [3]). Most biologically functional sites, especially protein binding sites, contain highly conserved positions where variation is restricted by the contacting protein. Therefore, tuples containing highly conserved positions are preselected. The second step maximizes the information content of the n-tuple. In the third step a fixed region around the tuple is included in C,maximization. This allows both detection of longer less conserved sites and discrimination of tuples found in 'random' locations which are not part of a binding site. Finally, CoreSearch sets the limits for C,maximization to the approximately biologically important region (determined by CoreSearch according to Frech et al., 1993).

Step 1: Selection of tuples containing highly conserved positions.

All basic tuples not containing highly conserved positions are eliminated for the reasons detailed above. The number of highly conserved positions, their relative positions within the tuple and the minimum percentage of the sequences in which these positions must occur are parameters. In the first step the positions of the best conserved nucleotides are determined for each basic tuple. Thereafter, a basic tuple is eliminated if less than the specified number of sequences fulfill this tuple's requirement for highly conserved nucleotides.

From all remaining basic tuples all members in the sequences not fulfilling this condition are eliminated unless this process would delete all members of the basic tuple in a sequence. In this case no tuple is eliminated for this sequence. This represents a first relaxation step which allows inclusion of a limited number of tuples varying from the preset conservation scheme. The limitation is that at least the preset threshold of tuple members complying with the conserved positions is present in the set of sequences before any relaxation is allowed.

Step 2: Maximization of tuple consensus index.

CoreSearch constructs tuple sets for each basic tuple (Fig. 1). A tuple set contains exactly one member of the basic tuple in each sequence (identical members at different positions are treated as one member at this point, see also Fig. 2). In order to limit the number of tuple sets to be analysed a suboptimal strategy was implemented (adapted from Stormo and Hartzell, 1989): Construction of tuple sets initiates with a subset of sequences comprising the lowest amount of tuple matches. Tuple sets are collected in an exhaustive manner by combining each match of each sequence with each match of the other sequences. This is done for a maximum number of sequences until the number of tuple sets created reaches an user defined threshold resulting in a number of tuple subsets. Starting with the sequences bearing the lowest amount of tuple matches allows inclusion of most sequences resulting in the best tuple subsets. Again in the order of least matches each match in the remaining sequences is combined with all existing tuple subsets and for each arising set the consensus index is calculated by the
Fig. 1. Overview over the CoreSearch selection steps. The small boxes represent tuples, the black dots above the boxes represent highly conserved positions. The hatched graph-inserts represent $C_i$ analyses. The longer bars indicate tuples including their surrounding sequences. The three steps are detailed in the Algorithm section of the text.

following formula (derived from Frech et al., 1993).

$$C_i = \frac{100}{(\log 4 \times n)} \times \sum_{i=0}^{n-1} \left( \log 4 + \sum_{b \in B} p(i, b) \times \log p(i, b) \right) \quad 0 \leq C_i \leq 100 \tag{3}$$

$n$ = tuple length

$p(i, b)$ = relative frequency of nucleotide $b$ at position $i$ of the $n$-tuples in the actual tuple set

$B = \{A, C, G, T\}$

For each original subset (exhaustive tuple combinations) only those combination(s) with a new tuple member which result in the maximum consensus index ($C_i$) are retained and used as new subsets in further analyses. All lower scoring combinations are discarded at this point. All remaining sequences are analysed in the same manner. In this way the exponential expansion of the number of tuple sets is limited. The whole procedure is repeated for all basic tuples. Finally CoreSearch eliminates all basic tuples for which the $C_i$ of the best tuple set is lower than a user defined percentage of the maximum tuple set $C_i$ of all basic tuples.

This prefers basic tuples with uneven distribution of mismatch location (see discussion). For each remaining basic tuple only a user defined number of best tuple sets is saved for further processing.
Identification of functional elements in unaligned nucleic acid sequences

Example:

The basic search algorithm defines the following 10 basic tuples: ATGTTTT, CTTGTTT, GTTTGCA, TGAGTCA, TTATGCA, TTGTTTT, TTTGCAA, TTTGCAC, TTTGTTT, TTTTTGC.

These basic tuples occur in most sequences more than once with different mismatches. The basic tuple TGAGTCA for example is contained in the sequences with no or one mismatch at the positions shown in Figure 2. There exist $10 \times 10 \times 10 \times 10 \times 10 = 10^5$ different tuple sets (At this point identical members are treated as a single member). If the maximum number of tuple sets for the optimal calculation method is set to 20 all combinations from the sequences with one match and the four first (Fig. 2 from top to bottom) of the six sequences with two matches will be built. This results in 16 basic tuple sets, two of which are shown in Figure 2.

The suboptimal method is applied only for the remaining three sequences starting with YSCHIS3, because it is the first remaining sequence (from top to bottom), which has only two matches. Each of its matches is added to each of the 16 basic tuple sets. For each of the 32 arising sets the $C_5$ is calculated and for each basic tuple set the best combination(s) is/are kept. If the two matches of YSCHIS3 never result in the same $C_5$ 16 tuple sets remain (Stormo and Hartzell, 1989). The process is repeated for SV40XX and finally for HPV-16, because this sequence has three matches. The five highest scoring tuple sets are:

<table>
<thead>
<tr>
<th>Tuple set</th>
<th>HPV-11</th>
<th>HPV-16</th>
<th>HPV-18</th>
<th>HPV-31</th>
<th>HPV-33</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPV-1</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
<tr>
<td>HPV-11</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
<tr>
<td>HPV-16</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
<tr>
<td>HPV-18</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
<tr>
<td>HPV-31</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
<tr>
<td>HPV-33</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
<td>TGTGCA</td>
</tr>
</tbody>
</table>

Figure 2. Tuple and position sets for AP-1 analysis (example). The figure shows all tuple members of the basic tuple TGAGTCA. Each box contains exactly one member. Identical tuples present at different positions of a single sequence represent one member as indicated by the boxes. Only two tuple sets (of a total of 192) and the four position sets of tuple set b are shown for clarity. Tuple members not belonging to set a or b are not boxed. For further details see text.
The whole calculation is done for all basic tuples. The maximum tuple set $C_i$ are as follows:

<table>
<thead>
<tr>
<th>Tuple set</th>
<th>ATGTTTT</th>
<th>CTGGTTT</th>
<th>GTTTGCA</th>
<th>TTTGTTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>75.73</td>
<td>76.46</td>
<td>73.50</td>
<td>79.16</td>
</tr>
</tbody>
</table>

The basic tuple TGAGTCA, which represents the biologically confirmed binding site scores best. If the minimum percentage of the maximum tuple set consensus index is set to 95% ($\Rightarrow C_i \geq 76.75$) the basic tuples ATGTTTT, CTGGTTT, GTTTGCA, TTTGTTT and TTTGCAC are eliminated.

**Step 3: Maximization of region consensus index.**

CoreSearch builds position sets for all existing tuple sets. Position sets contain exactly one member of the basic tuple in each sequence. In contrast to tuple sets identical members are distinguished by their position in the sequence (Fig. 2). This is necessary because a member from a tuple set can occur more than once in a sequence. Since the biologically correct positions of these identical tuple members cannot be identified solely from the information inside the tuple flanking bases must be taken into account in order to make the decision. Anyhow, flanking bases are an important criterion in further selection of the basic tuples. Therefore, position sets are created for each tuple set by the suboptimal strategy used for construction of the tuple sets (described in step 2). Each position set is used for an alignment employing the weight corrected alignment algorithm described in Frech et al. (1993). This corrects for the similarity of the sequences (a weight of 0.5 is assigned to each of two identical sequences which, therefore, are effectively treated as one sequence). The sum of $C_i$ values of the whole alignment region is calculated. From each original set only the top scoring combination is kept. This sum is called region consensus index ($RC_i$). This corresponds to step 3 in Figure 1.

**Example:**

Tuple set 5 shown above contains two tuples which occur twice in their sequence: TGAGTCA in PCGR at position 165 and 213 and TTAGTCA in SV40XX at position 152 and 224. Therefore, the following 4 position sets are obtained for this tuple set:

<table>
<thead>
<tr>
<th>Position set</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPV-1:</td>
<td>139</td>
<td>139</td>
<td>139</td>
<td>139</td>
</tr>
<tr>
<td>HPV-11:</td>
<td>303</td>
<td>303</td>
<td>303</td>
<td>303</td>
</tr>
<tr>
<td>HPV-16:</td>
<td>320</td>
<td>320</td>
<td>320</td>
<td>320</td>
</tr>
<tr>
<td>HPV-18:</td>
<td>188</td>
<td>188</td>
<td>188</td>
<td>188</td>
</tr>
<tr>
<td>HPV-31:</td>
<td>155</td>
<td>155</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td>HPV-33:</td>
<td>341</td>
<td>341</td>
<td>341</td>
<td>341</td>
</tr>
</tbody>
</table>

The whole procedure is repeated for every tuple set of all basic tuples. Finally, CoreSearch eliminates for each basic tuple all position sets with a $RC_i$ lower than a user defined percentage of the maximal $RC_i$ of the basic tuple. Furthermore the user can specify a maximum number of position sets for all basic tuples in order to limit the following time consuming analysis. CoreSearch eliminates position sets with the lowest $RC_i$ until the limit is reached. Basic tuples deprived of position sets are completely deleted.

**Example:**

The maximum $RC_i$ of the position sets of the remaining five basic tuples are listed below:

<table>
<thead>
<tr>
<th>Tuple set</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGAGTCA:</td>
<td>31.5</td>
</tr>
<tr>
<td>TTAGTCA:</td>
<td>28.1</td>
</tr>
<tr>
<td>ATGTTTT:</td>
<td>30.3</td>
</tr>
</tbody>
</table>

**Expansion of the position sets**

All position sets located so far are candidates for the basic representation of a potentially functional element based on the fairly strict condition of $n$-tuples with a single mismatch. This basic representation can be used to perform a more relaxed search in the sequences. For every position set a weight corrected analysis resulting in the delimitation of the potential binding site is carried out as described in Frech et al. (1993). Thereby, individual positions of each set are used as anchors for the alignment. Tuple members which do not match the extended consensus (up to 25 nucleotides) of all other members are rejected. This results in a complete consensus description for the binding site, which contains the relative frequency of each base respectively gap at each position. All sequences are now analysed for $n$-tuples which are similar to the $n$ positions in the consensus description to the right of the anchor point. The following formula is used to calculate a measure for the similarity between a $n$-tuple and the consensus core string.

$$\text{SIM} = \frac{\sum_{i=0}^{n-1} p(\text{anchor position } + i, \text{ tuple } (i))}{\sum_{i=0}^{n-1} \max(p(\text{anchor position } + i, b))}$$ (4)
Identification of functional elements in unaligned nucleic acid sequences

- anchor position = position of anchor in the consensus description
- p(pos,b) = relative frequency of base b at position pos in consensus description
- tuple(i) = base in tuple at position i
- \(\text{max}_{b \in B}(p(pos,b))\) = maximum relative frequency of a base b at position pos in the consensus description
- B = \{A,C,G,T\}

A similarity score of 1 represents the maximum possible similarity with this consensus core string. If the similarity score of a tuple exceeds an user defined threshold the position of the tuple in the sequence is used as anchor and the sequence is aligned with the consensus description. This alignment is compared to alignments of randomly shuffled versions of the sequence as described for the program ConsInspecetor in Frech et al. (1993). If the alignment is rated significant the position of the tuple is added to the position set used to create the consensus description provided the position has a distance of at least n nucleotides from an already defined position. Otherwise the tuples overlap and only the position scoring higher is retained. After all sequences are processed a second full analysis according to Frech et al. (1993) is carried out with the updated list of sequence positions.

This scheme is repeated for all remaining position sets. Only the consensus description containing the highest total sequence weight (i.e. more unrelated sequences) is selected if a basic tuple has more than one position set. This is based on the observation that the representation of a biological functional element is improved if it involves more unrelated sequences (Frech et al., 1993). Thus, CoreSearch creates one final consensus description for each retained basic tuple.

In our example (AP-1) CoreSearch requests 4 MB of RAM and <10 min. CPU time to complete the analysis.

Identification of multiple independent sequence elements

CoreSearch limits the number of tuple sets. Only tuple sets that reach a C, within the user defined range (% of best tuple set) are analysed. Thus, a high scoring tuple can suppress identification of additional lower scoring tuples in the same analysis.

Therefore, it is necessary to suppress elements already identified in the sequences in order to ensure identification of additional elements. The cores of all selected consensus descriptions are converted to upper case whereas the other sequence regions remain lower case. The modified sequence file can be used to initiate another CoreSearch analysis. The basic search algorithm described above is designed to ignore all nucleotides in upper case. This allows identification of additional non overlapping elements without bias towards previously found elements. Since new elements overlapping with the alignment region (usually ± 40 bp) of an identified core have already been merged with this core in the delimitation step mentioned above (see Frech et al., 1993) it is sufficient to identify one core of overlapping elements.

Results and discussion

The purpose of the described algorithm is to locate and define unknown potentially functional elements (e.g. protein binding sites) in sets of DNA sequences. This definition includes the position in each sequence, delimitation of the potentially important region and the base distribution at each position of the element. For the last two features we employed the ConsInd method which was shown to yield results closely correlated with biological data (Frech et al., 1993). However, this procedure requires the approximate position and the IUPAC core string of an element.

We have combined the new algorithm described in this report with the ConsInd algorithm (Frech et al., 1993) to directly create the description of a new unknown element (CoreSearch). If CoreSearch correctly analyses the sequence data it should be able to define the same elements as ConsInd from identical sets of sequences. Therefore, we analysed seven elements with known biological function for which a sufficient number of not too closely related sequences was available (for sequence names and accession numbers see Frech et al., 1993). The average sequence length was about 400 nucleotides. Table I shows the IUPAC consensus sequences of the regions (in capital letters) defined by ConsInd and CoreSearch. Consensus sequences (exact base distributions are used internally by both programs) are shown to facilitate visualization.

The results calculated by ConsInd were shown to be in agreement with experimental evidence from footprints, mutational studies or crystallography (Frech et al., 1993). The significant regions identified by CoreSearch agree almost perfectly with those calculated by ConsInd as shown in Table I. It is remarkable that CoreSearch is not incapacitated by highly homologous sequences as present in the SEF1 and U1A analyses. For example, the U1A sequences have averaged pairwise identities of 56% and the SEF1 sequences are 40% identical in average.

Since sequence lengths of about 500 bases in case of 7-tuples are allowed it seemed feasible to apply CoreSearch to a set of long terminal repeat (LTR) sequences. We chose LTR sequences since they are important regulatory structures of retroviruses and contain several well defined elements. Thus LTR sequences are suitable to test our programs ability to identify multiple elements in a single...
Table I. Comparison of the important regions (in capital letters) of biological functional sites identified by ConsFind and CoreSearch

<table>
<thead>
<tr>
<th>AP-I</th>
<th>ConsFind</th>
<th>wnn NTGASTCAn ynn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-I</td>
<td>CoreSearch</td>
<td>nnn MTGATGCAGN yy</td>
</tr>
<tr>
<td>NF-Y</td>
<td>ConsFind</td>
<td>nan TARCCAAATCAGN nrkc</td>
</tr>
<tr>
<td>NF-Y</td>
<td>CoreSearch</td>
<td>nnn KARCCAAATCAGN rnnn</td>
</tr>
<tr>
<td>TATA</td>
<td>ConsFind</td>
<td>nnnr NCTWTA AAN NnrY</td>
</tr>
<tr>
<td>TATA</td>
<td>CoreSearch</td>
<td>nnnnn CTATA AAAA Annnw</td>
</tr>
<tr>
<td>POLY A</td>
<td>ConsFind</td>
<td>nnt STTNTNAATAAGN ynt</td>
</tr>
<tr>
<td>POLY A</td>
<td>CoreSearch</td>
<td>nnnnnn NAATAAAAA Annnw</td>
</tr>
<tr>
<td>GRE</td>
<td>ConsFind</td>
<td>nny YTSKKNACAAANNTGTYCTNR gnn</td>
</tr>
<tr>
<td>GRE</td>
<td>CoreSearch</td>
<td>nyyyy kSKGTACMATSTGTCTCN gscn</td>
</tr>
<tr>
<td>SEFI</td>
<td>ConsFind</td>
<td>ncc AAACAGGTATCTGTTGTGY aknn</td>
</tr>
<tr>
<td>SEFI</td>
<td>CoreSearch</td>
<td>gcga AACAGGATATCTGTTGTGN annn</td>
</tr>
<tr>
<td>U1-A</td>
<td>ConsFind</td>
<td>yaa TTGCAC tctgggtgtgcwg ACCCCTG cr</td>
</tr>
<tr>
<td>U1-A</td>
<td>CoreSearch</td>
<td>ya TTGCAC tcyggrtgtgcwg ACCCCTG cr</td>
</tr>
</tbody>
</table>

The regions delimited by the programs are printed in capital letters while the surrounding sequences remained lower case. N represents a position which cannot be described by a two-nucleotide IUPAC code (W,S,M,K) but is still conserved to some extent.

set of sequences. We analysed 17 sequences with an average length of ~600 nucleotides. The parameters used for CoreSearch were the same as in the previous analyses. Figure 3 shows one of the sequences (AKV-LTR, Etzerodt et al., 1984; Herr, 1984) and known functional elements (adapted from Majors, 1990a).

Several runs of CoreSearch were necessary due to limitations of tuple numbers as detailed in the Algorithm section. CoreSearch first identified a core element CTTCTGT. The significant region is located between CCAAT and TATA box. The core element found overlaps with a CGCTT motif occurring three times between the CCAAT and TATA box. A second occurrence of this motif was found as additional peak as shown in Figure 3 in the analysis. These motifs are believed to contribute to promoter activity (Majors, 1990b). CoreSearch identified the core TATAAAA in the next run. The significant region exactly corresponds with the TATA box region mentioned above. In the third attempt CoreSearch identified the LVB element and the polyA downstream element (Frech et al., 1993; McLauchlan, 1985). Although the polyA signal was not found as primary element in the analysis it was clearly identified as additional peak in this run. In the next analysis an element comprising the end of the LVB site and the enhancer core was detected. The elements located in the third and fourth run occur in duplicate since they are located in the 99 nucleotide repeat of the LTR. The CCAAT box is detected in the fifth CoreSearch analysis of the sequences followed by two unknown elements in runs 6 and 8 and the identification of the GRE element in run 7. A ninth attempt of analysis yielded no further elements so CoreSearch analysis of these LTR sequences was completed with run 8.

CoreSearch requires some minimum conditions for successful analyses. The element must contain at least a core of about seven nucleotides differing in no more than two nucleotides from each other and at least seven independent sequences containing the element must be available. Fewer than seven sequences may prevent successful definition of the extension of the consensus element. Tuple length shorter than seven and/or more than one mismatch either impairs tuple selection or reduces the maximum sequence length that can be analysed. A 7-tuple search is successful for an average sequence length of up to 600 nucleotides.

Functional sites are expected to show a distinct distribution of the relative conservation of individual positions resulting in non random distribution of mismatches within the tuple. The consensus index of possible tuple sets of a basic tuple is very well suited to detect this type of conservation, since it assigns high values to highly conserved positions. A decrease of a high conservation results in a dramatic reduction of the assigned consensus index (C_i) while changes at less conserved positions have little effect on the C_i values. A tuple set from a basic tuple found by chance with the mismatch evenly distributed over all positions is therefore very unlikely to contain positions with a high C_i. Additional information about highly conserved positions can be utilized by the program if they are available from experimental data, but we show default values to work fine in all examples.

CoreSearch inevitably will find some additional random matches in some of the analysed sequences. This necessitates limitation of sequence lengths as detailed in the Algorithm section. A few additional random matches can be eliminated, since they are expected to differ from the distribution of mismatch positions present in the majority of individual representations of a basic tuple in the sequences. Analysis of the conservation of nucleotides surrounding a tuple set allows further elimination. Functional sites often exceed a 7-tuple, have a less conserved flanking region, and/or are closely distance correlated to other signals. On the other hand, tuple sets of a basic tuple found at random will probably also have a random surrounding region. Thus the mean C_i in the surrounding region of such a tuple set is expected to be lower than that of the region around a functional site. The same measure can be employed to distinguish between two identical matches of a basic tuple. The position sets with the highest region C_i most likely represent a biological function. Since the initial search uses a fairly strict condition and a position set contains one element per sequence we analyse the sequences for further elements similar to the found basic descriptions of the motifs. Here we use the methods described by Frech et al. (1993). This also ensures that the maximum information of the potential element can be utilized in
Identification of functional elements in unaligned nucleic acid sequences

The hallmark of CoreSearch is that absolutely no a priori knowledge about the sequence elements except their mere existence is required (as in some of the other programs). Our program allows modification of several parameters but did not require any changes of the default settings in any of the examples. Therefore, the user does not have to set parameters individually. CoreSearch differs in several respects from published methods. A small set of sequences is sufficient not only to detect the conserved core string but also to derive a complete description of the

**Fig. 3.** Sequence of the AKV-LTR (Etzerodt et al., 1984; Herr, 1984) with known functional elements and CoreSearch defined elements. Below the sequence known functional elements (derived from Majors, 1990a; Majors, 1990b) are indicated by black bars and their common identifiers. The boxes represent the regions determined by CoreSearch as significant element candidates and the numbers in circles correspond to the number of the CoreSearch run in which the elements were defined. Boxes with a black triangle on top represent additional peaks identified during the CoreSearch analyses. Elements found in runs 6 and 8 are currently under experimental investigation and therefore not shown. The analysis was exhaustive with run 8. A ninth run did not reveal any more elements.

Contrast to the limited information present in the initial tuple.
complete region with potential biological function. This sets CoreSearch apart from the methods described in (Bucher and Bryan, 1984; Galas and Waterman, 1985; Mengeritsky and Smith, 1987) which do require a priori knowledge about the elements (in the form of prealigned sequences). Furthermore, these methods are impractical for tuples longer than six nucleotides. The method described in Pesole et al. (1992) is restricted to words (tuples) tolerating no mismatch and does not discriminate multiple occurrences of the same pattern within one sequence which is important for any element consisting of more nucleotides than the defined pattern.

Though the methods described in (Lawrence and Reilly, 1990; Cardon and Stormo, 1992) can analyse sequences without a priori knowledge this would require numerous exploratory models as the authors state. Our method carries out the complete analysis automatically and no user interaction is required during the whole process after the initial dialog.

CoreSearch works best with a relatively small number of sequences. We cannot analyse more than ~20–30 sequences in reasonable time on a common workstation. However, this is no real shortcoming since more than 20 experimentally verified binding sequences for an unknown factor are rarely available. We have already shown that 15–20 sequences are sufficient to establish a consensus description which is only marginally improved by the addition of more sequences (Frech et al., submitted). This description can then be used to locate additional matches of this motif in an unlimited amount of sequence data (Conslndector, Frech et al., 1993).

Another important feature of CoreSearch is its ability to detect multiple independent elements in one set of sequence data. The case of complete LTR sequences emphasizes the advantage of the inclusion of features from the program Conslnndector. Although CoreSearch does not find all elements (e.g. the polyA signal) as core strings the delimitation algorithm identifies further significant regions in addition to core strings on the basis of a distance correlation to other elements.

In conclusion CoreSearch should be a useful method complementing the existing tools for definition and detection of transcriptional elements in nucleotide sequences. Since it does not require any a priori knowledge about the elements to be analysed and can successfully analyse small sets of sequences it should be especially useful for the analysis of new sequence data that have been selected by functional assays (e.g. for protein binding) from a random pool of sequences or clone libraries. Results can be immediately used with our freely available program Conslndector to find candidate elements in new genomic sequence data suitable for experimental verification.

Acknowledgements

The authors would like to thank Dr R. Brack-Werner, Kerstin Quandt, and Burkhard Morgenstern for critical reading of the manuscript. Part of this work was supported by grant 413-4001-01-IV-BE/B 281-28 of the BMBF.

References


