Simultaneous alignment and annotation of *cis*-regulatory regions.

A. S. Bais[1], S. Grossmann[1], M. Vingron[1]

Abstract

Genomic transcriptional processes rely heavily on the combinatorial binding of transcription factors on the upstream regions of genes. Analysis and complete characterization of such 'regulatory' regions can unlock the door to the understanding of regulatory and transcriptional mechanisms. Such analysis is usually based on the prediction of transcription factor binding site (TFBS) hits on non-coding regions. To improve the prediction of TFBS hits, phylogenetic-footprinting is generally employed. Here we propose a method that combines sequence comparison and scanning with TFBS hits to generate simultaneously annotated alignments. The hope is that simultaneous alignment and annotation can result in an improved specificity.

1 Introduction

Positional-specific scoring matrices (PSSMs) and phylogenetic footprinting combined together have been the backbone of recent research on identifying *cis*-regulatory elements in non-coding regions around genes. Most approaches based on these methods carry out two individual steps sequentially. Extract the conserved regions by generating alignments between the two sequences and scan these extracted regions either individually or together for TFBS hits. The order of the two steps could also be reversed.
In this paper, we present an approach that carries out the alignment while simultaneously annotating with TFBS hits. The PSSMs of interest are incorporated explicitly into the alignment model, hence extending the usual description of alignments and alignment scoring schemes. Equipped with such an *extended* alignment model, we can generate not only the locally optimal (as is usually done) but also the suboptimal alignments, all with TFBS hit annotations.

1.1 Basic Idea

As input, the method takes in a pair of non-coding sequences that surround orthologous genes in two species at a reasonable evolutionary distance. At the core is a set of *profiles* denoted by $\mathrm{P}_1, \ldots, \mathrm{P}_q$, representing motifs one wishes to detect. The final outcome is a set of non-intersecting, locally optimal alignments of the concerned sequences, with additional annotations of profile instances on the generated alignments. In order to generate this additional annotation, we need to extend the usual definition of gapped local alignments.
For a standard gapped alignment, it suffices to keep track of the aligned positions (*substitutions*) and the insertions and deletions (*indels*). Calculation of the score of the alignment, then, requires summing up the substitution scores as evaluated from entries of a substitution matrix $\mathtt{s}$ and then subtracting a gap-penalty g for every indel. For two given sequences, an optimal local alignment can then be determined by using the Smith-Waterman [2] algorithm, based on dynamic programming. For non-overlapping suboptimal alignments a simple implementation could be based on the Waterman-Eggert algorithm[3].
To include profiles for our method, we extend the usual scoring scheme as follows. Let l_k be the length of a profile P_k. Now, any stretch of l_k consecutive non-gapped substitutions in the alignment can be scored in (at least) two ways: either as in the standard gapped alignment by scoring each of the l_k letter pairs using the scoring matrix $\mathtt{s}$ entries, or as a profile instance using a *profile-scoring array* $\mathtt{PSA}_k$. The $\mathtt{PSA}_k$ reflects how well the l_k letter pairs fit to the motif described

[1]Dept. of Computational Biology, Max Planck Institute for Molecular Genetics, Berlin, Germany, E-mail: `bais@molgen.mpg.de`

by P_k. Thus, $\texttt{PSA}_k$ is a function which assigns a real-valued score to every pair of non-gapped sequences of length l_k.
With the usual scoring scheme thus extended, we are then capable of determining the different kinds of optimal local alignments for two give sequences x and y and a set of profile-scoring arrays $\texttt{PSA}_1, \dots, \texttt{PSA}_q$ by slight modifications of the standard algorithms (discussed later).

2 Notations and Definitions

We introduce some notations for standard alignments and then proceed to explain how the proposed alignment model can be viewed as an extension of the standard model.

2.1 Standard alignments

Let Σ be a finite alphabet and $|\Sigma|$ its cardinality. Although we are primarily concerned with DNA sequences ($|\Sigma| = 4$), we stick to a more general exposition here.
Let $x = (x_1, x_2, \dots, x_m)$ and $y = (y_1, y_2, \dots, y_n)$, $x_i, y_j \in \Sigma$ be two sequences of lengths m and n, respectively. Let $\texttt{s}\colon \Sigma \times \Sigma \to \mathbb{R}$ be the substitution scoring matrix. Let $g > 0$ be the gap penalty. A gapped alignment A between x and y introduces gaps into the two sequences such that the lengths of the resulting two sequences are identical, and places these gapped sequences one upon the other so that no gap is ever above another gap. Columns with letters in both sequences are *substitutions* (and denoted by S), and are scored using entries from $\texttt{s}$, whereas every column which contains a gap in either x or y is an *insertion* (I) or *deletion* (D), respectively, and a gap-penalty g is subtracted for it. Hence, an alignment A can be coded as a string of letters $\{\mathrm{S}, \mathrm{D}, \mathrm{I}\}$ which represents the sequence of its columns. As an example, the two sequences $x =$ ACGTATAACG and $y =$ ACCATATATC could be aligned as following,

$$\begin{array}{rccccccccccc} x^*\colon & \mathrm{A} & \mathrm{C} & - & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{T} & \mathrm{A} & \mathrm{A} & \mathrm{C} & \mathrm{G} \\ y^*\colon & \mathrm{A} & \mathrm{C} & \mathrm{C} & \mathrm{A} & \mathrm{T} & \mathrm{A} & \mathrm{T} & \mathrm{A} & \mathrm{T} & \mathrm{C} & - \\ \alpha\colon & \mathrm{S} & \mathrm{S} & \mathrm{I} & \mathrm{S} & \mathrm{S} & \mathrm{S} & \mathrm{S} & \mathrm{S} & \mathrm{S} & \mathrm{S} & \mathrm{D} \end{array}$$

We denote by $n_{\mathrm{S}}(A)$, $n_{\mathrm{D}}(A)$ and $n_{\mathrm{I}}(A)$, the respective numbers of these three letters in that string. Therefore, $A = (\alpha_1, \alpha_2, \dots, \alpha_{h(A)}) \in \{\mathrm{S}, \mathrm{D}, \mathrm{I}\}^{h(A)}$ is a valid alignment, of length $h(A)$, between x and y, iff $n_{\mathrm{S}}(A) + n_{\mathrm{D}}(A) = m$ and $n_{\mathrm{S}}(A) + n_{\mathrm{I}}(A) = n$. An alignment with the above characteristics directly defines a mapping

$$\begin{aligned} A\colon \Sigma^m \times \Sigma^n &\to (\Sigma^*)^{h(A)} \times (\Sigma^*)^{h(A)} \\ (x, y) &\mapsto (x^*, y^*), \end{aligned}$$

where $\Sigma^* := \Sigma \cup \{-\}$ and $h(A) = n_{\mathrm{S}}(A) + n_{\mathrm{D}}(A) + n_{\mathrm{I}}(A)$. The score $S(A, x^*, y^*)$ for the complete alignment is calculated by summing over the indvidual column scores and the alignment with the highest score amongst all possible alignments is the optimal (global) alignment between x and y. For every column i, the scoring scheme defines a score $S_c(\alpha_i, x_i^*, y_i^*)$ via

$$S_c(\alpha_i, x_i^*, y_i^*) = \begin{cases} \texttt{s}(x_i^*, y_i^*), & \text{if } \alpha_i = \mathrm{S} \\ -g, & \text{if } \alpha_i = \mathrm{D}, \mathrm{I} \end{cases}$$

The score for the complete alignment is simply calculated as the sum of the column scores, $S(A, x^*, y^*) = \sum_{i=1}^{h(A)} S_c(\alpha_i, x_i^*, y_i^*)$.

2.2 Extended alignments

The proposed notion of extended alignments can be conveniently introduced on the string coding representation of alignments discussed above. Recall that our aim is to directly assign some parts of the alignment to one of the q profiles $\mathrm{P}_1, \dots, \mathrm{P}_q$ with lengths $l_1, \dots, l_q$, respectively. Therefore, we now code an alignment A by a string of letters from the set $\{\mathrm{S},\mathrm{D},\mathrm{I},\mathrm{P}_1,\dots,\mathrm{P}_q\}$. In this coding, a column of A coded by P_k means that the corresponding l_k letters from x and y are gaplessly aligned and assigned to profile P_k. Therefore, in the previous example with the two sequences x and y, a possible alignment could be,

$$\begin{array}{rcccccccc} x^*: & \mathrm{A} & \mathrm{C} & - & \mathrm{G} & \mathrm{TATAA} & \mathrm{C} & \mathrm{G} \\ y^*: & \mathrm{A} & \mathrm{C} & \mathrm{C} & \mathrm{A} & \mathrm{TATAT} & \mathrm{C} & - \\ \alpha: & \mathrm{S} & \mathrm{S} & \mathrm{I} & \mathrm{S} & \mathrm{P}_1 & \mathrm{S} & \mathrm{D} \end{array}$$

In continuation of our former notation, we denote the number of occurences of letter P_k in A to be $n_{\mathrm{P}_k}(A)$. Therefore, A can be a valid alignment for x and y if $n_\mathrm{S}(A)+n_\mathrm{D}(A)+len_k(A)=m$ and $n_\mathrm{S}(A)+n_\mathrm{I}(A)+len_k(A)=n$, where $len_k(A)=\sum_{k=1}^{q} l_k \times n_{\mathrm{P}_k}(A)$, and $h(A)=n_\mathrm{S}(A)+n_\mathrm{I}(A)+n_\mathrm{D}(A)+\sum_{k=1}^{q} n_{\mathrm{P}_k}(A)$.

For the mapping $(x,y)\mapsto(x^*,y^*)$, we have in the extended setting

$$x_i^*, y_i^* \in \begin{cases} \Sigma^* & \text{if } \alpha_i \in \{\mathrm{S},\mathrm{D},\mathrm{I}\} \\ \Sigma^{l_k} & \text{if } \alpha_i = \mathrm{P}_k,\ 1 \le k \le q. \end{cases}$$

Coming to the scoring, the column score S_c is now extended as

$$S_c(\alpha_i, x_i^*, y_i^*) = \begin{cases} \mathtt{s}(x_i^*, y_i^*), & \text{if } \alpha_i = \mathrm{S} \\ -g, & \text{if } \alpha_i = \mathrm{D},\mathrm{I} \\ \mathtt{PSA}_k(x_i^*, y_i^*) - p_k, & \text{if } \alpha_i = \mathrm{P}_k,\ 1 \le k \le q \end{cases}$$

where $\mathtt{PSA}_k$ is the *profile scoring array* associated with profile P_k and p_k is an additional calibration parameter (discussed later). The definition given in the previous section for the score of the complete alignment pertains.

3 Algorithm

Since we are interested in regions with high local similarities, we use the Smith-Waterman algorithm for generating the optimal local alignments, as our foundation step and extend it for incorporating profiles.

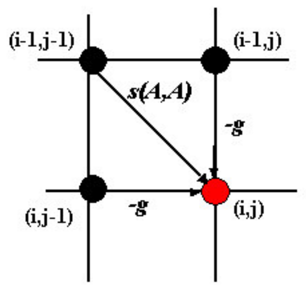
Smith-Waterman Algorithm Let $\mathtt{M}$ be the $(m+1)\times(n+1)$ dynamic programming matrix which is filled up with the following recursion rule:

$$\mathtt{M}(i,j) = \max\{0, \mathtt{M}(i-1,j-1)+\mathtt{s}(x_i,y_j), \mathtt{M}(i-1,j)-\mathrm{g}, \mathtt{M}(i,j-1)-\mathrm{g}\} \quad (1)$$

That is, $\mathtt{M}(i,j)$ is the maximum of 0 or the best score of an alignment ending at x_i and y_j. Depending on whether it is derived from $(i-1,j-1)$, $(i-1,j)$ or $(i,j-1)$, $\mathtt{M}(i,j)$ is called a substitution (S), deletion (D) or an insertion (I), respectively. Thus, in the usual scenario, the score of a grid point is calculated using the scores of three predecessor points.(Fig. 1). The alignment is then found by tracing back from the point (i,j) with the maximum score $\mathtt{M}(i,j)=\mathtt{M}_{\max}$, where $\mathtt{M}_{\max}=\max_{k,l}\mathtt{M}(k,l)$, $\forall\ 0\le k\le m,\ 0\le l\le n$.

Proposed Algorithm The recursion step described above, is modified to incorporate profiles as follows. Besides the three predecessors considered before, we now also compare the score at a point l_k diagonal positions above the point (i,j), that is, the score of the point (i',j'), where $i'=i-l_k+1$, $j'=j-l_k+1$.(Fig. 2). For each profile, we introduce a new $m\times n$ matrix, $\mathtt{pmat}_k$, defined as

$$\mathtt{pmat}_k(x_i,y_j) = \begin{cases} -\infty & \forall\ i,j \le l_k \\ \sum_{t=1}^{l_k} \mathtt{PSA}_k(x_{i-l_k+t}, y_{j-l_k+t}, t) - p_k & \text{otherwise} \end{cases}$$



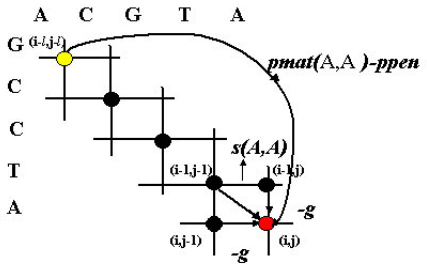


Figure 1: Recursion step in the Smith-Waterman algorithm.

Figure 2: Modified recursion step in the proposed algorithm where $l_k = 5$.

Here, $1 \leq t \leq l_k$ is simply a variable introduced for simplicity of presentation. Hence, each $\mathtt{pmat}_k$ entry is derived from $\mathtt{PSA}_k$ by summing over previous l_k values. The pair-profile hits are accomodated by considering the profile scores at each point in the recursion step,

$$\mathtt{M}(i,j) = \max \begin{cases} 0, \\ \mathtt{M}(i-1, j-1) + \mathtt{s}(x_i, y_j), \\ \mathtt{M}(i-1, j) - \mathrm{g}, \\ \mathtt{M}(i, j-1) - \mathrm{g}, \\ \mathtt{M}(i-(l_k-1), j-(l_k-1)) + \mathtt{pmat}_k(x_{i-1}, y_{j-1}) \\ \ \forall\, 1 \leq k \leq q \end{cases}$$

Hence, $\mathtt{M}(i,j)$ is the maximum of 0 or the best score of an alignment with pair-profile hits ending at x_i and y_j.As before, we traceback from the point with the maximum score, $\mathtt{M}(i,j) = \mathtt{M}_{\max} = \max_{k,l} \mathtt{M}(k,l)$, for all $0 \leq k \leq m$, $0 \leq l \leq n$.

4 Suboptimal Alignments

We discuss a straightforward approach for generating suboptimal alignments with pair-profile hits based on the Waterman-Eggert algorithm [3]. During the first traceback, any point (i,j) in the alignment that has been arrived at from a substitution *or* a pair-profile hit is marked as "used", accordingly. Any point that is not an indel or lies inside a pair-profile jump is not marked.
The procedure is then on the same lines as the Waterman-Eggert[3] algorithm. During recalculation, the same algorithm is carried out again, but with one difference. At every step in the alignment, the substitution or pair-profile instance with which the alignment ended previously is not allowed again. That is, for any point which was marked "used" as a substitution, the recursion step now changes to disallow the substitution again, meaning, it does not consider the immediate diagonally above point. For a point which was marked "used" as a pair-profile end, the recursion step disallows the same profile hit to occur here again. So, if $\mathtt{M}^*$ is the re-computed matrix, to calculate $\mathtt{M}^*(i,j)$, we consider the column (k,j), where $i < k$. Then each entry (k,j), $k = i+1, i+2, \ldots$ is re-computed until the new value $\mathtt{M}^*(k,j) = \mathtt{M}(k,j)$ using the modified recursion rules. A similar calculation for the row (i,k), $j = k+1, k+2, \ldots$ is carried out until $\mathtt{M}^*(k,j) = \mathtt{M}(k,j)$. Traceback starts as usual from the maximum-score element.
We now explain in detail how one could make a balanced choice of the scoring elements of the extended scheme.

5 Scoring issues

As mentioned before, the standard alignment scoring scheme has two main constituents: the scoring matrix $\mathtt{s}$ and the linear gap-penalty g. The entries of the scoring matrix $\mathtt{s}$ can usually be interpreted as log-likelihood ratios with respect to some background letter distribution π. Let $q = (q(u,v))_{u,v\in\Sigma}$ be a matrix giving the probability of observing letter pair (u,v) at an aligned position. Hence,

$$\mathtt{s}(u,v) = \mathrm{c}_{\mathtt{s}} \log\left(\frac{q(u,v)}{\pi(u)\pi(v)}\right)$$

The constant $\mathrm{c}_{\mathtt{s}}$ is a scaling constant introduced for computational convenience. The two new parameters introduced in the extended model are the profile-scoring array, $\mathtt{PSA}_k$ and the *profile-penalty*, p_k. Both together need to maintain a balance between two things: the individual profiles against the standard alignment scores and the scores of the profiles amongst themselves, so that none get unduly over- or under-represented in the optimal alignment. We assume in the following that a background letter distribution π and a scaled scoring matrix $\mathtt{s}$, which properly reflects the evolutionary distance between the two sequences, have been given. Also, each profile is assumed to be given via its *position specific letter distributions*, or formally $\mathrm{P}_k = (\mathrm{P}_k^1, \dots, \mathrm{P}_k^{l_k})$, where each P_k^i is a probability distribution on Σ.

Profile-scoring array Let $\mathbf{u}, \mathbf{v} \in \Sigma^{l_k}$. $\mathtt{PSA}_k$ will be constructed as a scaled and rounded log-odds score of generating $\mathbf{u}$ and $\mathbf{v}$ from two independent samples from P_k as opposed to sampling all their letters from independent calls to π. That is,

$$\frac{\mathrm{P}_k(\mathbf{u})\mathrm{P}_k(\mathbf{v})}{\prod_{i=1}^{l_k}\pi(u_i)\pi(v_i)} = \prod_{i=1}^{l_k} \frac{\mathrm{P}_k^i(u_i)}{\pi(u_i)} \frac{\mathrm{P}_k^i(v_i)}{\pi(v_i)}$$

which after taking logarithms, re-ordering and scaling leads to,

$$\begin{aligned} \mathtt{PSA}_k(\mathbf{u},\mathbf{v}) &:= \sum_{i=1}^{l_k} \mathrm{c}_{\mathtt{s}} \log\left(\frac{\mathrm{P}_k^i(u_i)\mathrm{P}_k^i(v_i)}{\pi(u_i)\pi(v_i)}\right) \\ &= \sum_{i=1}^{l_k} \mathrm{c}_{\mathtt{s}} \log\left(\frac{\mathrm{P}_k^i(u_i)}{\pi(u_i)}\right) + \sum_{i=1}^{l_k} \mathrm{c}_{\mathtt{s}} \log\left(\frac{\mathrm{P}_k^i(v_i)}{\pi(v_i)}\right) \\ &=: \mathtt{PSSM}_k(\mathbf{u}) + \mathtt{PSSM}_k(\mathbf{v}). \end{aligned} \tag{2}$$

where, c_s is a scaling constant, introduced for convenience. Here, $\mathtt{PSSM}_k$ is the usual (scaled) *position specific scoring matrix* as it would be constructed from P_k for a single sequence and a given background letter distribution.
The special form (2) which states that the profile score of the string pair $(\mathbf{u},\mathbf{v})$ can be written as the sum of the individual PSSM scores of the two strings, comes from the fact that we have assumed the generation of $\mathbf{u}$ and $\mathbf{v}$ to be independent in the underlying probabilistic model. It is also possible, in our approach, to explicitly model the evolutionary conservation in the string pair $(\mathbf{u},\mathbf{v})$ but this will, of course, destroy the special additive form in (2). Such a modeling should respect the evolutionary distance between the two species under consideration, which also has (implicitly) been used in the construction of the scoring matrix $\mathtt{s}$. The following discussions on the balancing of different scoring parameters are not affected by such a more general construction of profile scoring arrays.

Profile penalty Let $\mathbf{u},\mathbf{v}$ be two substrings of length l_k of the given sequences x and y, respectively. Recall (Section 2) that for every alignment which gaplessly aligns u, v by successively visiting l_k substitution (S) states, there exists a variant which uses a P_k state for $\mathbf{u}$ and $\mathbf{v}$, instead. In a direct comparison of the two alignments, the latter achieves a higher score than the former, whenever $\mathtt{PSA}_k(\mathbf{u},\mathbf{v}) - p_k > \sum_{i=1}^{l_k} \mathtt{s}(u_i, v_i)$. which, after rewriting, is equivalent to

$$\mathrm{LLR}_{\mathrm{P}_k^2, q^{l_k}}(\mathbf{u},\mathbf{v}) := \log \frac{\mathrm{P}_k(\mathbf{u})\mathrm{P}_k(\mathbf{v})}{\prod_{i=1}^{l_k} q(u_i,v_i)} > \frac{p_k}{\mathrm{c}_{\mathtt{s}}} =: p_k'. \tag{3}$$

Therefore, we can see that the decision about which of the two variants of the alignment achieves a higher score is nothing but a log-likelihood ratio test. Two models for the generation of $(\mathbf{u}, \mathbf{v})$ are compared. One model (the null-hypothesis) samples the letter pairs in $(\mathbf{u}, \mathbf{v})$ independently from the evolutionary letter pair distribution q, whereas the other model (the alternative) generates u and v independently from the model defined by P_k.
The choice of p'_k can therefore be based on the same kind of considerations which can be used in every log-likelihood ratio based test. For example, for a given level α one could determine $p'_k(\alpha)$ such that $\mathbb{P}_{q^{l_k}}(\mathrm{LLR}_{\mathrm{P}_k^2, q^{l_k}}(\mathbf{u}, \mathbf{v}) > p'_k(\alpha)) < \alpha$. In the testing language this would mean that a test, which rejects the null-hypothesis in favour of the alternative, whenever the test statistics $\mathrm{LLR}_{\mathrm{P}_k^2, q^{l_k}}(\mathbf{u}, \mathbf{v})$ exceeds the value $p'_k(\alpha)$ has a *type-I-error* below α.
A further measure which could be considered when specifying a threshold p'_k is the *power* of the test, or $\beta(p'_k) := \mathbb{P}_{\mathrm{P}_k^2}(\mathrm{LLR}_{\mathrm{P}_k^2, q^{l_k}}(\mathbf{u}, \mathbf{v}) > p'_k)$. A more detailed discussion about how to use these two types of errors when specifying a profile penalty p'_k can be found in [1].

6 Conclusion and Outlook

We have presented a method that simultaneously aligns and annotates *cis*-regulatory regions. We have shown how the traditional dynamic-programming method can be extended straightforwardly to incorporate profiles for the proposed model. The scoring parameter selection has also been discussed. The algorithms discussed have been implemented and are currently under evaluation.

References

[1] S. Rahmann, T. Müller, and M. Vingron. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2(1):Article 7, 2003.

[2] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology.*, 147:195–197, 1981.

[3] M. S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *Journal of Molecular Biology*, 197:723–728, 1987.