Assessing the Discordance of Multiple Sequence Alignments

Amol Prakash and Martin Tompa

Abstract—Multiple sequence alignments have wide applicability in many areas of computational biology, including comparative genomics, functional annotation of proteins, gene finding, and modeling evolutionary processes. Because of the computational difficulty of multiple sequence alignment and the availability of numerous tools, it is critical to be able to assess the reliability of multiple alignments. We present a tool called StatSigMA to assess whether multiple alignments of nucleotide or amino acid sequences are contaminated with one or more unrelated sequences. There are numerous applications for which StatSigMA can be used. Two such applications are to distinguish homologous sequences from nonhomologous ones and to compare alignments produced by various multiple alignment tools. We present examples of both types of applications.

Index Terms—Multiple sequence alignment, discordance, alignment accuracy, Karlin-Altschul statistics, biology and genetics, life and medical sciences, computer applications.

1 Background

Over the last two decades, multiple sequence alignments have contributed to our understanding of molecular biology, whether through comparison of homologous genomic regions or analysis of protein families based on sequence similarity. However, multiple sequence alignment is a difficult computational problem [46]. Many different heuristic tools are available, and these tools often produce incomparable results. Because of this, a number of recent reviews and articles [3], [17], [20], [21] have made compelling pleas for methods to assess the accuracy of multiple sequence alignments and to compare the alignments produced by different tools.

There is a large body of work to try to address this important problem, particularly in the context of multiple protein alignment:

1. One approach to measuring the accuracy of multiple sequence alignment tools is to use artificial sequences resulting from a simulation of evolutionary processes [5], [25], [26], [31], [42]. Since the experimenter can track all evolutionary events, identifying the truly homologous characters is straightforward. One can then align the simulated sequences using any multiple sequence alignment tool and use the known homologies to measure the accuracy of this alignment. An obvious drawback of this simulation approach is its sensitivity to assumptions about the underlying evolutionary processes, which are not at all well understood.
2. Another approach is to run the alignment program on a set of sequences in which certain features are known a priori to be homologous and measure the accuracy with which these known homologous features are aligned. This approach was used in genome-size alignment studies by Brudno et al. [6], Margulies et al. [20], and Wang et al. [45]. The most obvious choice for the known homologous features is a set of coding exons. However, this choice suffers from the shortcoming that such features are usually well conserved and easy to align, and most algorithms do so quite accurately. In addition to using coding exons, Margulies et al. [20] also tested alignment accuracy using ancestral repeats, and Wang et al. [45], using noncoding RNA, both of which tend to be more challenging to align correctly than coding exons. For this general approach though, many known sets of homologous sequences have been discovered using some alignment algorithm, which leads to circularity if then used to test the accuracy of an alignment algorithm.
3. A number of papers have suggested methods that inspect multiple sequence alignments, judging regions of the alignment that show good conservation across the aligned sequences to be well aligned and even removing sequences from the alignment that show lack of good conservation [8], [23], [43], [44]. While good conservation often implies good alignment, the converse need not be true. It is certainly the case that perfectly aligned sequences will vary greatly in conservation from column to column and region to region, rendering conservation a questionable predictor of alignment quality.
4. Lassmann and Sonnhammer [18] proposed a measure to assess alignment quality by comparing

- *A. Prakash is with the Biomarker Research Initiative in Mass Spectrometry (BRIMS) Center, Thermo, 790 Memorial Drive, Suite 201, Cambridge, MA 02139. E-mail: amol.prakash@thermofisher.com.*
- *M. Tompa is with the Department of Computer Science and Engineering, Box 352350, University of Washington, Seattle, WA 98195-2350. E-mail: tompa@cs.washington.edu.*

Manuscript received 12 July 2007; revised 10 Oct. 2007; accepted 18 Nov. 2007; published online 5 Dec. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2007-07-0080. Digital Object Identifier no. 10.1109/TCBB.2007.70271.

several multiple sequence alignments, assuming that regions identically aligned by multiple tools are more reliable than regions differently aligned. This method requires several auxiliary alignments in order to evaluate the alignment of interest. In the same general class are methods that employ other auxiliary information such as the protein secondary structure in order to assess alignment quality [11].

We pursue a statistical approach quite different from any of these, by extending the theory of Karlin and Altschul [14], [15] from pairwise to multiple sequence alignments. Our method can be used to assess the reliability of any given multiple sequence alignment, which is not true of the methods in categories 1 and 2 above.

1.1 A Statistical Theory of Sequence Relatedness

In the case of pairwise alignment, the most common method to assess relatedness (functional or evolutionary) of a pair of sequences is by aligning them and then testing the null hypothesis that there would be an equally good local alignment in a random pair of unrelated sequences. Karlin and Altschul [14] introduced statistics for this purpose that were later employed in the popular tool BLAST [2]. Unfortunately, for the case of more than a few sequences, there has been little progress in extending such an assessment to multiple sequence alignments. Prakash and Tompa [29] presented initial ideas to extend the BLAST statistics to multiple alignments. The main challenges were to develop a realistic null hypothesis and an appropriate score function. Both these problems and their solutions are described in detail in that paper and also summarized next.

The motivating problem is to identify when a multiple sequence alignment is contaminated with unrelated sequences. To build a scalable yet robust methodology, we assume that there is an unrooted phylogenetic tree relating the sequences. Our null hypothesis consists of those cases where a single phylogeny branch (whose removal partitions the sequences into two disjoint subsets) exhibits "unrelated behavior," that is, the two subsets are each homologous within themselves but independent of each other. For each such branch of the phylogeny, we compute the p-value of the score of the pairwise alignment of the two disjoint subsets of the multiple alignment. This tests the null hypothesis that the two subsets are independent. Finally, the assumption made is that when we reject all the cases of the null hypothesis (one case for each branch of the phylogeny), this is sufficient evidence that the multiple alignment shows all the sequences to be related. There are two justifications for this assumption. First, the cases of unrelated sequences that are hardest to detect are the ones where the error occurs on only one branch of the phylogeny, the remaining subalignments being correct. Second, since most multiple alignment tools follow a progressive alignment strategy over a phylogeny, such errors also seem the most common. Thus, by this approach, we anticipate capturing the most common and the hardest cases. See Fig. 1 (explained in Section 3.1) for the experimental justification of this assumption.

In this paper, we bring together these ideas in a tool called Statistical Significance of Multiple Alignments (StatSigMA) that can be used to assess whether multiple

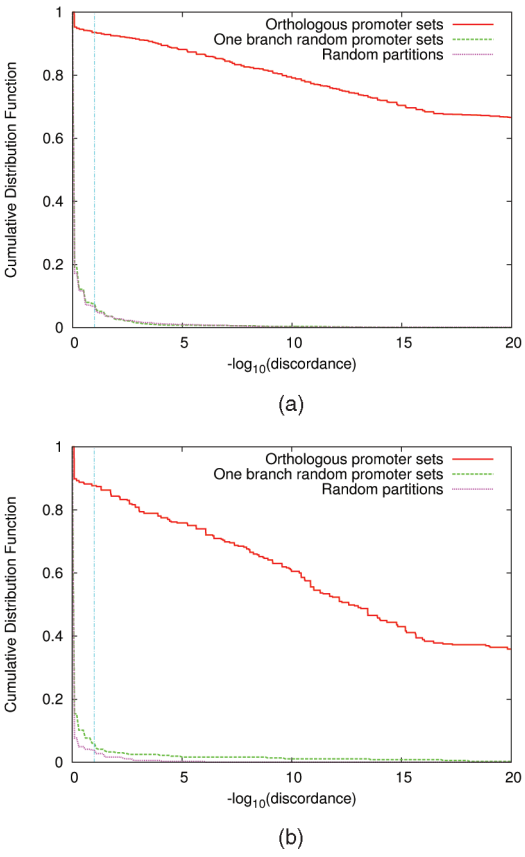

Fig. 1. Cumulative distribution function of the discordances computed by StatSigMA for many (a) human/chimp/mouse/rat and (b) human/chimp/mouse/rat/chicken promoter sets. (a) Graphs are plotted for the 4,215 orthologous promoter sets, 5,000 random promoter sets having unrelated behavior on a single branch of the phylogeny, and 5,000 promoter sets created using random partitions of the species. The curves for the latter two types of promoter sets are so close that they are difficult to distinguish. (b) Graphs are plotted for the 777 orthologous promoter sets, 5,000 random promoter sets having unrelated behavior on a single branch of the phylogeny, and 5,000 promoter sets created using random partitions of the species.

alignments of nucleotide or amino acid sequences are contaminated with one or more unrelated sequences. StatSigMA provides biologists with a principled way to assess the relatedness of sequences in multiple sequence alignments produced by any alignment tool. Just as BLAST E-values provide a measure for ranking pairwise alignments, StatSigMA's "discordance" scores provide a measure for ranking multiple alignments. We describe changes that have been made to this method since its introduction [29] to make it more sound and accurate both in theory and practice and also to make it more efficient. We present new results using StatSigMA to compare the output of various multiple alignment tools and to assess the reliability of protein alignments. The applicability to protein multiple alignments is new: the earlier method [29] was too inefficient for use in proteins.

StatSigMA measures the reliability of a multiple alignment by its highest scoring collection of columns (see Section 2). This is analogous to using BLAST to find a significant match to a query sequence, since BLAST also reports the significance of the best local alignment (or collection of local alignments) between query and subject.

Even in the case of pairwise alignment, there is a theory of local alignment significance but no known theory of global alignment significance. StatSigMA builds on this theory of local alignment significance.

We have recently extended the methods described here in order to apply them to whole-genome multiple alignments [30], which is a more complex problem. The method described in this paper is intended to be used on relatively short multiple sequence alignments (for instance, protein alignments or promoter alignments); it produces a single "discordance" value that measures how well aligned the best collection of alignment columns is, much as BLAST produces a single E-value for a pairwise alignment that measures how related the best collection of alignment columns is. In contrast, our recent extension to whole-genome multiple alignments [30] assigns such a discordance value to every region of the alignment, so that the user can see which regions are trustworthy and which ones are less so. That extension relies in many places on the technical details presented here, such as the detailed p-value formulas (see Section 2), which are not repeated in the extension paper.

1.2 Alignment Correctness versus Conservation

Many methodologies have been suggested for distinguishing conserved regions from nonconserved ones, for example, binCons [19], phastCons [36], GERP [9], and Gumby [27]. Each of these takes a multiple alignment and identifies well-conserved regions in it, *assuming that the alignment is correct*. In contrast, StatSigMA assesses the reliability of the multiple alignment itself. This should be applied to any multiple alignment before any of the tools listed above are applied to identify well-conserved elements.

Some researchers have proposed using conservation thresholds as a surrogate for filtering out poorly aligned regions. As mentioned earlier, while good conservation often implies good alignment, the converse certainly need not be true. Conservation measures are designed to detect regions under purifying selection; those evolving at neutral rates will exhibit much lower conservation scores, even though they may be perfectly aligned. To make matters worse, good conservation need not even imply good alignment: Prakash and Tompa [30] have shown that misalignment of one sequence can often be found even in regions where conservation scores are very high due to strong conservation in the remaining sequences.

Other existing alignment scoring methods (such as sum of pairs or percent identity) are also measures of sequence conservation, though in these cases, there is not even a phylogeny upon which to model evolutionary conservation.

2 Implementation

StatSigMA takes as input a multiple sequence alignment. The sequences in the multiple alignment are used to infer a phylogeny relating those sequences. (Alternatively, the user can provide a phylogeny.) The algorithm followed by StatSigMA is outlined below. Each of the steps is described and discussed in detail later.

Input: Multiple sequence alignment.

1. Create a phylogeny from the sequence data (unless the user has supplied a phylogeny).
2. For every branch k of the phylogeny do the following steps:
   a. Create the scoring function corresponding to unrelated behavior on branch k.
   b. Using the scoring function, estimate Karlin-Altschul parameters K_k, λ_k, and H_k [14].
   c. Using the scoring function, identify the maximal scoring segments of the input multiple alignment.
   d. Using the Karlin-Altschul parameters K_k, λ_k, and H_k identify the set of maximal scoring segments resulting in the least p-value p_k.
3. Output $\max_k p_k$ as the *discordance*.

Analogous to a p-value, the discordance is between zero and one, and the lower its value, the better the alignment (in the sense of not being contaminated with unrelated sequences).

2.1 Creating a Phylogeny

As described earlier, our approach is based on the assumption that there is an unrooted phylogeny relating the sequences of the multiple alignment. Each case of the null hypothesis corresponds to unrelated behavior on a single branch k of that phylogeny. The term "unrelated behavior" refers to independent sequences that should not be aligned, as follows: The removal of branch k separates the phylogeny into two subtrees t_1 and t_2. The null hypothesis assumption is that the sequences at the leaves of t_1 are related according to the multiple alignment, as are the sequences at the leaves of t_2, but these two subalignments are independent of each other. (Another way to state the null hypothesis would be to treat the branch exhibiting unrelated behavior as if it were of infinite length. This is a more traditional way of formulating the null hypothesis in molecular phylogenetics.) The assumption we make then is that rejecting all the single-branch null-hypothesis cases is sufficient evidence that the multiple alignment shows all the sequences to be related. The justifications for this assumption were presented above.

If the user does not supply a phylogeny, we use the phylogeny generated by MUSCLE [10]. While neighbor-joining trees [33] are closer to the real evolutionary trees, we use the default option of UPGMA trees [38] in MUSCLE, as they are closer to what most alignment tools use in practice.

Note that methods have been proposed previously for comparing two sequence profiles. If t_1 and t_2 are the two subtrees separated by the removal of the branch k that exhibits unrelated behavior, we could have constructed profiles from each of the two sets of sequences at the leaves of t_1 and t_2, respectively, and then applied Karlin-Altschul statistics to these two profiles to test whether or not they should be aligned together. However, such profiles necessarily ignore the underlying phylogeny. What we will do instead in the next sections is to apply Karlin-Altschul statistics to scores based on an evolutionary model that respects the phylogeny.

2.2 Scoring Function

Suppose we have S sequences in a multiple alignment of length N related by a phylogenetic tree T (having branch lengths). Let $\gamma_1, \gamma_2, \ldots, \gamma_S$ be the residues observed at a particular column of the multiple alignment. Suppose we want to test the hypothesis that there is unrelated behavior on branch k. Suppose the removal of branch k separates T into subtrees t_1 (having residues $\beta_1, \beta_2, \ldots, \beta_i$ at the leaves) and t_2 (having residues $\beta_{i+1}, \beta_{i+2}, \ldots, \beta_S$ at the leaves). Let M be the evolutionary model. Then, analogous to the Karlin-Altschul log-likelihood score [14], the score for observing this column of the multiple alignment is given as follows:

$$sc_k(\gamma_1, \ldots, \gamma_S | T, M) = \log\left(\frac{\Pr(\gamma_1, \ldots, \gamma_S | T, M)}{\Pr(\beta_1, \ldots, \beta_i | t_1, M)\Pr(\beta_{i+1}, \ldots, \beta_S | t_2, M)}\right). \quad (1)$$

We precompute this score for all possible tuples at the leaves of the tree. If the alphabet size is α, this requires precomputing and storing α^S scores. This is infeasible when either α is large (e.g., for proteins) or when there is a large number of sequences. In such cases, we precompute the scores for only the tuples present in t_1 and t_2 in the multiple alignment. If the two subalignments each have length N, then N^2 tuples can be formed by aligning the N tuples from either of the subtrees with each other. We assume that these N^2 tuples constitute a good sample of the background. In the case when the alignment length is very small, we add a pseudocount (1/N) to all integral scores to correct for the small sample size. Estimating the p-value in this way, by sampling from the sequence itself, is known as composition-based statistics [35], [34] and has been shown to be more sensitive than a standard background distribution. In either case, we precompute only $\min(N^2, \alpha^S)$ scores.

The various probability terms in (1) are computed using the dynamic programming algorithm of Felsenstein [12]. In our evolutionary model, gaps are treated as single-character deletions. Masked sites, incomplete information about some sites, and unaligned sites (e.g., in TBA [5] alignments) are treated as characters drawn randomly from the alphabet with background probabilities. These characters were called ϵ by Prakash and Tompa [29]. For the tree T, the score of a tuple containing ϵ is the same as the score of the tuple over a tree with the leaf having ϵ removed.

Treating gaps as single-character deletions penalizes long gaps heavily. We chose this approach as it gives us a first handle on gaps for multiple sequences. Proper handling of gaps remains a hard problem to solve even for pairwise alignment statistics [24].

2.3 Estimating Karlin-Altschul Parameters

Karlin and Altschul [14] gave approximation methods to compute the parameters K, λ, and H, given the probability distribution of the scores. The parameters K and λ can be thought of simply as natural scales for the search space size and the scoring system, respectively. The parameter H accounts for edge effects. Using the precomputed scores, and the probabilities of seeing those scores in independent sequences ($\Pr(\beta_1, \ldots, \beta_i \mid t_1, M) \times \Pr(\beta_{i+1}, \ldots, \beta_S \mid t_2, M)$ in (1)), and the code to compute the Karlin-Altschul parameters (provided by Stephen Altschul), we estimate the parameters.

The methods suggested by Karlin and Altschul [14] require integral scores. Equation (1) outputs real numbers. Using a large multiplier for these scores, followed by conversion to integers, leads to small rounding errors. As suggested by Schäffer et al. [34], a larger multiplier results in more accurate parameters, but the time complexity of the methods used to estimate the Karlin-Altschul parameters is cubic in the value of the multiplier. Thus, a large multiplier slows down the computation significantly. The code provided by Stephen Altschul uses a vector to store the background probabilities of the various scores (all integral scores between the minimum and maximum scores). The length of this vector is proportional to the multiplier used. Instead, we implement this vector as a sparse list. This provides significant savings in computation time, thus making the use of a large multiplier feasible. We use a multiplier of 1,000, which results in negligible rounding errors. A dense score vector (which can be the result of a large tree, a complex evolutionary model, varying background probabilities for the various residues, etc.) can slow down this process considerably.

2.4 Identifying Maximal Scoring Segments

Once the scoring function has been precomputed, we can give a score to each column of the multiple alignment. Any contiguous set of columns in the multiple alignment is uniquely identified by a starting and an ending column. The score of such a set of contiguous columns is just the sum of the scores of its individual columns. We use the algorithm of Ruzzo and Tompa [32] to identify all maximal positively scoring segments in this alignment. This is a linear-time algorithm that finds all the best nonoverlapping contiguous sets of columns having positive scores, where the kth best segment is defined recursively to be the one that maximizes its score among all segments disjoint from the $k-1$ best segments.

Summing column scores corresponds to assuming that the columns of the multiple alignment are independent. While this assumption is unrealistic, there is no good understanding of how to avoid this assumption even in the pairwise case. Thus, tools such as BLAST make the same assumption.

2.5 P-Value for a Single Branch

Once we have estimated the parameters K_k, λ_k, and H_k corresponding to the branch k, we can compute the p-value of the score using the sum statistics of Karlin and Altschul [15] as follows: Let $sc_{k,1}, sc_{k,2}, \ldots, sc_{k,r}$ be the scores of the r highest scoring nonoverlapping segments. Let $sc'_{k,1}, sc'_{k,2}, \ldots, sc'_{k,r}$ be the respective normalized scores, where for the ith segment

$$sc'_{k,i} = \lambda_k sc_{k,i} - \ln\left(K_k(N - H_k)^2\right), \quad (2)$$

where N is the length of the alignment.

Define $total_{k,r} = (\sum_{i=1}^{r} sc'_{k,i}) - ln(r!)$. Then, the p-value of $total_{k,r}$ for the null-hypothesis case k is given as follows [15]:

$$p\text{-value}(z_{k,r}|k,r) = \Pr(total_{k,r} \geq z_{k,r}|k,r) = \int_{z_{k,r}}^{\infty} \frac{e^{-t}}{r!(r-2)!}\left(\int_0^{\infty} y^{r-2} exp\left(-e^{(y-t)/r}\right)dy\right)dt. \quad (3)$$

The expression in (3) is precomputed using MATLAB for the score range -100 to 100 and for a maximum of 50 segments. This result is then used to compute the p-value of $total_{k,r}$.

For deciding the best value of r, we choose the value that results in the least p-value. For this, we consider segments in decreasing order of their scores. We continue including segments as long as we observe a decrease in the Bonferroni-corrected p-value. Multiple-hypothesis correction (Bonferroni correction) is performed using the ideas of Altschul [1], and thus, we multiply the p-value of $total_{k,r}$ by 2^r to give a conservative estimate of the p-value:

$$p\text{-value}(z_{k,1}, z_{k,2}, \ldots|k) = \min_r \left(p\text{-value}(z_{k,r}|k,r) \times 2^r\right). \quad (4)$$

This choice of r corresponds to the most significant collection of segments.

2.6 Discordance for the Entire Tree

Once we have computed the p-value of the score for every branch of the phylogeny, we report the maximum p-value among all branches as the *discordance* of the multiple alignment. This corresponds to the p-value of the weakest branch. The idea behind taking the maximum p-value is that even if one branch has a p-value greater than the least level of significance at which the null hypothesis is rejected, the null hypothesis should not be rejected.

2.7 Improvements in StatSigMA

As mentioned earlier, the implementation described above contains several improvements to the method since its introduction [29] that make it more accurate and efficient. This section summarizes those changes.

Perhaps the greatest improvement is the method now used to estimate the Karlin-Altschul parameters. In the previous version, these were estimated by simulation, which in retrospect was inaccurate and inefficient, as further described in Section 3. We now use the hill-climbing method proposed by Karlin and Altschul [14]. This change makes the estimates not only much more accurate but also now efficient enough to allow the application of StatSigMA to protein alignments, which was infeasible in the earlier version. The large multiplier and sparse vector implementation discussed earlier for estimating these parameters is also new, as is the incorporation of the edge effect parameter H_k.

The earlier version gave the discordance for the whole tree as the average, rather than the maximum, of the single-branch p-values; the new method is more sound. The use of the Ruzzo-Tompa algorithm [32] for identifying maximal scoring segments is new. The multiple hypothesis correction for single-branch p-values is also new.

2.8 Time Complexity

To estimate the parameters, we first need to build the scoring function. Suppose there are S sequences in the multiple alignment. Let N be the length of the multiple alignment and α be the alphabet size. As described in Section 2.2, we precompute $\min(N^2, \alpha^S)$ scores. Each such computation requires computing the log likelihood ratio in (1) using the algorithm by Felsenstein [12]. This dynamic programming step takes time $O(\alpha)$ for every node of the tree T and for each possible residue at the node. Thus, computing one score takes time $O(S\alpha^2)$, and the total time required to compute the scoring function is $O(S\alpha^2 \min(N^2, \alpha^S))$.

After computing the scoring function, the estimation of parameters is also time consuming, and it is dependent on the density of the sparse score vector (described in Section 2.3). Let the time taken to estimate the parameters be T_{param}. Once the parameters have been computed, identifying the maximal scoring segments takes time $O(N)$. We repeat the entire process for all branches, that is, $O(S)$ iterations. Thus, the time complexity of StatSigMA is $O(S^2\alpha^2 \min(N^2, \alpha^S) + ST_{param} + SN)$.

To give a few instances of execution times observed in practice, StatSigMA takes 2 seconds on an alignment of four DNA sequences, each of length 1,000, and 2-3 minutes on an alignment of nine DNA sequences, each of length 1,000. For protein sequences, an alignment of five sequences each of length 50 requires 1-2 minutes, and an alignment of five sequences each of length 400 requires about 25-30 minutes. Thus, this tool slows down considerably for protein alignments. Less than 5 percent of the time is spent in estimating the parameters (ST_{param} in the time complexity given in the previous paragraph). All experiments were run on 2.6-GHz Intel Xeons with Linux as the platform.

3 Results

3.1 Comparative DNA Sequence Benchmark

In another study [28], we collected large sets of high-confidence orthologous promoter regions from human, chimp, mouse, rat, and chicken. This was done by collecting orthologous genes and filtering out those that did not have identifiably orthologous transcription start sites. The upstream sequences were masked for repeats using RepeatMasker [37] and DUST [39]. This left us with 4,215 sets of orthologous promoter regions from human, chimp, mouse, and rat and 777 promoter sets from human, chimp, mouse, rat, and chicken. For each of these, we aligned the length-1,000 sequences upstream of the transcription start sites using MUSCLE [10] and then computed the discordance of each alignment using StatSigMA. Fig. 1a plots the cumulative distribution function of the discordances for the human/chimp/mouse/rat promoter sets, and Fig. 1b plots the cumulative distribution function of the discordances for the human/chimp/mouse/rat/chicken promoter sets.

Using these promoter sets, we also created 5,000 promoter sets having one randomly chosen branch of the phylogeny exhibiting unrelated behavior, for instance, a promoter set having orthologous sequences from the two primates and orthologous sequences from the two rodents, but these two pairs taken from two randomly and

independently chosen promoter sets. We also created 5,000 promoter sets based on random partitions of the species, for instance, orthologous human and mouse sequences, but chimp and rat sequences taken from promoter sets chosen randomly and independently of each other and of the chosen human-mouse promoter set. Similar random promoter sets were created from the promoter sets that include chicken. The sequences in each of these random promoter sets were also aligned using MUSCLE [10], and then, StatSigMA was used to compute the alignment's discordance. The cumulative distribution function of the discordances of the alignments of these sets is also plotted in Fig. 1.

As can be seen in Fig. 1, StatSigMA can clearly distinguish homologous promoter sets from ones that are contaminated with unrelated sequences. Using a discordance threshold such as 0.1 (the vertical line in Fig. 1), we have significant alignments for more than 90 percent of the orthologous human/chimp/mouse/rat promoter sets, more than 85 percent of the orthologous promoter sets including chicken, and approximately 8 percent-9 percent of one-branch random promoter sets. As for the one-branch random promoter sets, we should expect 10 percent of the promoter sets to have a discordance less than 0.1 and 1 percent to have a discordance less than 0.01, because this data fits our null hypothesis. The plot in Fig. 1 shows a good fit to these points. The plot for random partitions validates our assumption of approximating the superexponential number of possible partitions by the linear number of one-branch partitions [29]. The discordances for random partitions are slightly greater (that is, the alignments look slightly worse) than the ones for one-branch random promoter sets.

Comparing Fig. 1 with the plot for the same promoter sets analyzed earlier [29], we see substantial differences, particularly in the one-branch random graphs: see Fig. 2. In the earlier study [29], we estimated the Karlin-Altschul parameters using simulation, via TBA [5] alignments on a large number of one-branch random promoter sets. In these simulations, TBA often produced poor alignments, which is not surprising as these are nonhomologous promoter sets. Thus, the Karlin-Altschul parameters estimated were inaccurate, causing subsequent discordance calculations to be inaccurate. In fact, as the parameters were based on poor alignments, when we used these incorrect parameters for alignments over one-branch random promoter sets, we severely underestimated the p-values. In the current work, we use the hill-climbing methods suggested by Karlin and Altschul [14] to estimate the parameters rather than simulation, obtaining much more accurate (and efficient) estimates. This is one of the major improvements that StatSigMA incorporates.

3.2 Comparison of Multiple Alignment Tools

The next result shows the application of StatSigMA to compare multiple alignment tools. We took the orthologous promoter sets described in the previous section (with and without chicken) and compared the discordances of ClustalW [7] and TBA [5] alignments. ClustalW is a classical global multiple alignment tool, using a progressive alignment based on the algorithm of Needleman and Wunsch [22]. TBA is a more modern local alignment method that uses an anchor-based approach. Very highly conserved regions are first identified as anchors and then extended to produce an alignment between these anchors. To be fair to the alignment tools, we considered only those promoter sets that have at least 700 unmasked residues. This left us with 2,134 promoter sets for human/chimp/mouse/rat and 364 promoter sets including chicken.

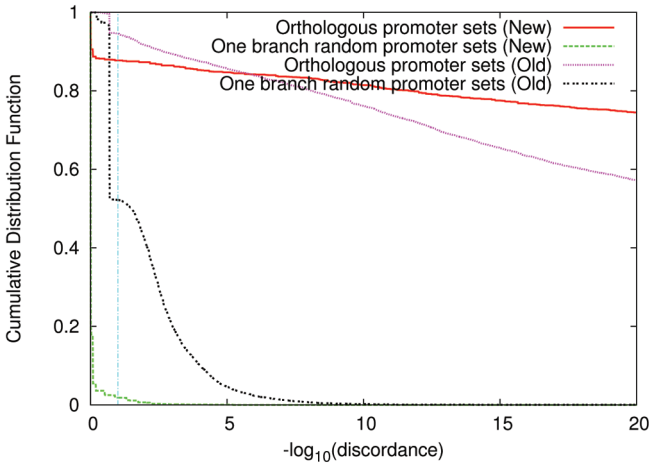

Fig. 2. Cumulative distribution function of the discordances computed by StatSigMA for orthologous human/chimp/mouse/rat promoter sets and for random promoter sets having unrelated behavior on a single branch of the phylogeny. Graphs are shown both for the old method of estimating Karlin-Altschul parameters by simulation and for the new method by hill climbing. For consistency with the earlier study [29], all alignments in this figure use TBA rather than MUSCLE. This explains small discrepancies between Fig. 1a curves and those labeled "New" here.

It may seem at first that StatSigMA is not the appropriate tool to compare two different alignments of the same sequences. After all, StatSigMA identifies sequences that are unrelated to the remainder, and the alignments being compared both involve the same set of sequences. However, when one or more sequences are badly misaligned, StatSigMA will identify them as not belonging. This is entirely analogous to the Karlin-Altschul statistics for comparing two pairwise alignments of the same sequences: if one of them is badly misaligned, these statistics will assign it a much greater E-value.

For each promoter set, we computed the log ratio of 1) StatSigMA's discordance for the alignment produced by TBA and 2) StatSigMA's discordance for the alignment produced by ClustalW. Fig. 3 plots the histogram of these log ratios. The plot for human/chimp/mouse/rat shows that for most promoter sets, both ClustalW and TBA produce similar quality alignments, but for about 20 percent of the promoter sets, ClustalW produces much less discordant alignments. This includes 10 percent of the promoter sets having a log ratio greater than 20. For another 20 percent of the promoter sets, TBA produces less discordant alignments. The plot for human/chimp/mouse/rat/chicken, however, shows that ClustalW produces less discordant alignments on more than 60 percent of the promoter sets, which was unexpected.

Analyzing these alignments in detail, we found that there are many instances when TBA fails to find good

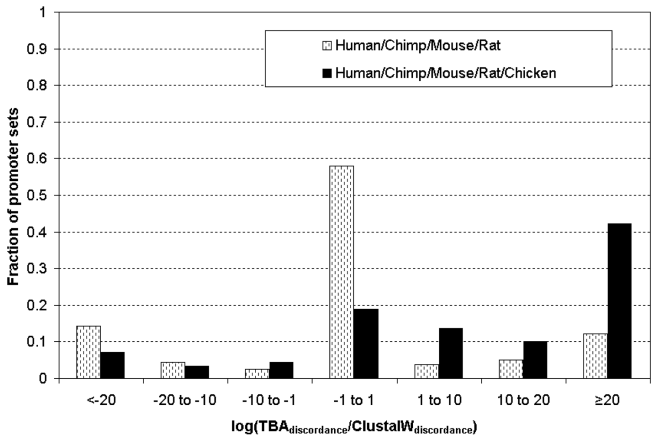

Fig. 3. Discordances of ClustalW and TBA alignments for 2,134 orthologous human/chimp/mouse/rat and 364 orthologous human/chimp/mouse/rat/chicken promoter sets are computed using StatSigMA. The histogram plots $\log_{10}$ (discordance(TBA)/discordance(ClustalW)) against the fraction of promoter sets.

anchors to start with, especially in chicken. Thus, it fails to align those promoter sets, thereby producing alignments with discordance 1. As we have a lot of trust in the orthology of these promoter sets (Prakash and Tompa [28] and Fig. 1b), we expect to see a good alignment. ClustalW, which is not anchor based and always produces a global alignment including all species, is able to find those good alignments. As chicken is quite distant from the mammals, on more than 40 percent of the promoter sets, TBA failed to find anchors to start with, whereas ClustalW produced a low-discordance alignment (log ratio greater than 20). Analyzing the promoter sets where TBA did better than ClustalW, we find that these promoter sets are such that the alignment is very skewed: for example, the beginning part of the human sequence aligns with the end part of the mouse sequence. ClustalW, trying to produce a global alignment, tends to produce a discordant one on these promoter sets. These results show that different multiple alignment tools have their weaknesses and strengths, and StatSigMA provides an unbiased measure of which tool produces the better alignment. It is possible that by using a different set of parameters for TBA, we may be able to find anchors in more promoter sets and thus produce less discordant alignments, but then, we need a method to evaluate the various alignments produced at different choices of parameters. This could also be done using StatSigMA.

A point to note is that our result differs from previous studies that showed TBA to perform better than ClustalW [5], [28]. Blanchette et al. [5] were using simulated data among mammals. Prakash and Tompa [28] were evaluating alignments for the purposes of identifying small well-conserved regulatory elements. In both scenarios, TBA would be able to find anchors to start with (the well-conserved elements) and thus produce a good alignment. This result is consistent with our findings. But we also report that there are many other real orthologous data sets that do not have small strongly conserved elements. On these, ClustalW performs better than TBA. Pollard et al. [25] obtained results similar to ours. They showed that global alignment tools such as ClustalW have a higher sensitivity on the entire sequence, but the local alignment tools have a higher specificity.

In addition, one should notice that the comparison in Fig. 3 is based on the insistence that the alignment include all species. For instance, it is quite possible that TBA produces a very reliable alignment of the mammals but refuses to align chicken, for which it would be penalized in this comparison. This points out that there are various possible criteria one could use for such a comparison.

In this work, we have shown the comparison of only two multiple alignment tools. A much more systematic comparison of many more tools on a variety of data sets using StatSigMA is planned for the future. As a note of caution, StatSigMA's assessment is of local multiple alignments. Currently, the statistics of global alignments are not well understood even for the pairwise case. Therefore, when we compare the various multiple alignment tools, we are comparing them by the quality of the local alignments they produce. For global alignment tools such as ClustalW, we find the set of best induced local alignments and then compare to TBA's local alignment. Our method should not be applied beyond this, that is, comparing various global alignment tools for the best global alignment is not something that can be achieved by StatSigMA. As most studies use multiple alignment tools to create an alignment and then study locally well-conserved regions, we believe that StatSigMA provides a very useful (and the first) statistical methodology to compare various tools for this purpose.

3.3 Protein Alignments

The next result shows the performance of StatSigMA on protein alignments. BAliBASE (version 2.0) [41] is a database of manually refined protein multiple sequence alignments specifically designed for the evaluation and comparison of multiple sequence alignment programs. It consists of databases categorized by sequence length and similarity. Fig. 4 plots the cumulative distribution function of the discordances computed by StatSigMA for the alignments of three of these databases (referred to as Ref1 in BAliBASE).

The sequences in each of these protein sets are equidistant, that is, the percent identity between two sequences is within a specified range, and the sequences are of similar length with no large insertions or extensions. The sequence identity among the various Ref1 protein sets varies from less than 25 percent to more than 35 percent. The short alignments have a length of approximately 50 amino acids, the medium alignments have a length of 100-200 amino acids, and the long alignments have a length of 300-400 amino acids. As shown in Fig. 4, long alignments are less discordant than short ones for similar percent identity. This is not surprising, since long alignments are more likely than short alignments to have some higher scoring segment or multiple high scoring segments.

The figure also plots the cumulative distribution of the discordances for one-branch random protein sets. For this, we took a BAliBASE protein set, calculated the phylogeny using MUSCLE [10], picked a branch, and replaced the subtree on one side of the branch by a similar-sized subtree

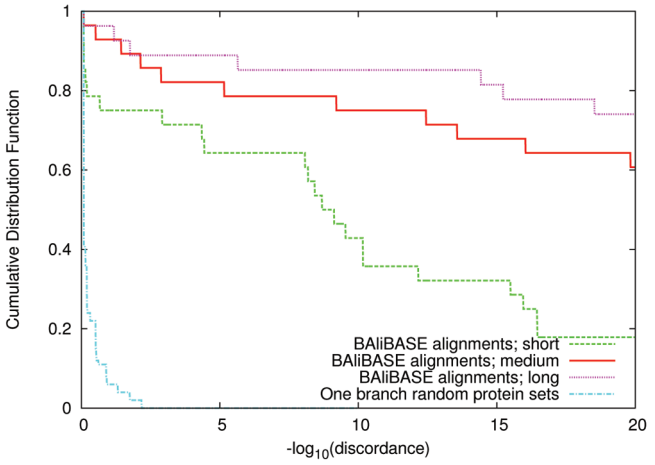

Fig. 4. Cumulative distribution function of the discordances computed by StatSigMA for BAliBASE protein alignments (Ref1), for various protein sets having different lengths. A graph is also plotted for 1,000 protein sets artificially created having a single unrelated branch.

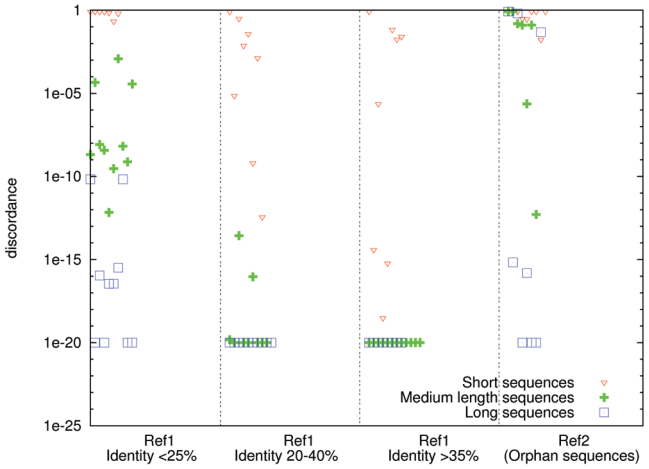

Fig. 5. Scatter plot for discordances of various BAliBASE protein sets; with varying percent identities (Ref1). Scatter plots are also shown for Ref2, which includes some orphan (very low similarity) protein sequences along with the sequences from a single family. In each case, the protein sets are further classified into sets having short sequences, medium-length sequences, and long sequences. The discordances are thresholded at 10^{-20}.

chosen from a random BAliBASE protein set. The leaf sequences were then realigned using MUSCLE, and the discordance of this alignment was computed using StatSigMA. The resulting cumulative distribution is similar to the ones in Fig. 1. Fig. 4 shows that StatSigMA can clearly distinguish true protein families from a mixture of two families, even for short proteins. Note that this is true even though StatSigMA measures the alignment discordance by its maximal scoring local segments alone.

We also plot the discordances of the BAliBASE protein sets as a scatter plot. Fig. 5 plots the discordances of various BAliBASE Ref1 protein sets with varying lengths and percent identities. Another type of protein set (referred to as Ref2 in BaliBASE) aligns up to three *orphan* sequences (less than 25 percent identical) from Ref1 with a family of at least 15 closely related sequences. As expected, a higher percent identity and longer sequences both result in less discordant alignments. The discordances of Ref2 alignments are on the higher side, and this is due to Ref2 incorporating orphan sequences into a family of closely related sequences. This suggests that there is sometimes not enough information in an alignment of the primary sequences alone to decide whether some of the sequences are unrelated. This is consistent with the construction of BAliBASE [41], where structural information was used to create the alignments.

4 Conclusions

In this work, we presented some results that involved assessing the discordance of multiple alignments using StatSigMA. We believe that StatSigMA can be used for many other applications:

1. StatSigMA can be used to identify homologous regions in various genomes. As demonstrated in Fig. 1, StatSigMA can distinguish between homologous promoter regions and an alignment generated from unrelated sequences. A filtering-based solution was presented for this problem by Prakash and Tompa [28].
2. A related project in progress is to use StatSigMA to evaluate the quality of the whole-genome alignments available on the UCSC [16] and Ensembl [4] Web browsers. ENCODE [40] regions make an interesting data set for this problem as well, as there are more than two dozen species sequenced for these regions and multiple alignments of these sequences by four different methods [20]. We have preliminary results [30] on the UCSC 17-vertebrate alignment using StatSigMA.
3. StatSigMA can be used to compare the alignments produced by various multiple alignment tools. In Fig. 3, we presented the results of comparing ClustalW and TBA on the basis of the alignments that they produce for high-quality homologous promoter regions. The same ideas can be extended to compare the performance of various tools and various choices of parameters (of the multiple alignment tools) on any set of sequences.
4. StatSigMA can also be used to assess the quality of databases such as HomoloGene [47] that use multiple alignment similarity to infer functional or ancestral relationship. Figs. 3 and 4 showed the result of a similar test performed on BAliBASE.
5. Tools such as TBA [5] and PROTONET [13] produce a multiple alignment in some form or another in their output. In the process of calculating their result, these tools make decisions about when to merge two multiple alignments into a single alignment. The decision made is based on thresholds whose values impact the output significantly. Ideas from StatSigMA can be used to choose such thresholds in a principled manner, thus giving an accurate estimate of the quality of the result produced.

Incorporating a more realistic gap model is a future need for StatSigMA. The current model treats gaps as single-character deletions, which penalizes long gaps very heavily. Incorporating an affine gap model for multiple alignments should improve StatSigMA's accuracy. Doing so requires a detailed understanding of the evolution of gaps that no one yet has.

As discussed earlier, StatSigMA is currently slow to run on alignments involving long or many protein sequences. Both of these are very interesting cases, so improving StatSigMA's runtime on such cases is desirable.

5 Availability

StatSigMA is freely available for download at http://bio.cs.washington.edu/software.html. It has been implemented in C++ under the GNU license on a Linux platform.

Acknowledgments

The authors thank Stephen Altschul, Ewan Birney, Mathieu Blanchette, Jonathan Carlson, Joe Felsenstein, Phil Green, Nan Li, Michal Linial, Webb Miller, Larry Ruzzo, Saurabh Sinha, Rosalia Tungaraza, Zasha Weinberg, Zizhen Yao, the Ensembl Help Desk, the UCSC Help Desk, the Univerity of Washington Genome Sciences support staff, and the anonymous reviewers for their support, suggestions, and feedback. This material is based upon work supported in part by the US National Science Foundation under Grant DBI-0218798 and by the US National Institutes of Health under Grant R01 HG02602.

References

[1] S. Altschul, "Evaluating the Statistical Significance of Multiple Distinct Local Alignments," *Theoretical and Computational Methods in Genome Research,* S. Suhai, ed., pp. 1-14, 1997.

[2] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic Local Alignment Search Tool," *J. Molecular Biology,* vol. 215, pp. 403-410, 1990.

[3] S. Batzoglou, "The Many Faces of Sequence Alignment," *Briefings in Bioinformatics,* vol. 6, no. 1, pp. 6-22, 2005.

[4] E. Birney, T.D. Andrews, P. Bevan, M. Caccamo, Y. Chen, L. Clarke, G. Coates, J. Cuff, V. Curwen, T. Cutts et al., "An Overview of Ensembl," *Genome Research,* vol. 14, pp. 925-928, 2004.

[5] M. Blanchette, W.J. Kent, C. Riemer, L. Elnitski, A.F. Smit, K.M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E.D. Green, D. Haussler, and W. Miller, "Aligning Multiple Genomic Sequences with the Threaded Blockset Aligner," *Genome Research,* vol. 14, no. 4, pp. 708-715, Apr. 2004.

[6] M. Brudno, C. Do, G. Cooper, M.F. Kim, E. Davydov, E.D. Green, A. Sidow, and S. Batzoglou, "LAGAN and Multi-LAGAN: Efficient Tools for Large-Scale Multiple Alignment of Genomic DNA," *Genome Research,* vol. 13, no. 4, pp. 721-731, 2003.

[7] R. Chenna, H. Sugawara, T. Koike, R. Lopez, T.J. Gibson, D.G. Higgins, and J.D. Thompson, "Multiple Sequence Alignment with the Clustal Series of Programs," *Nucleic Acids Research,* vol. 31, pp. 3497-3500, 2003.

[8] M. Cline, R. Hughey, and K. Karplus, "Predicting Reliable Regions in Protein Sequence Alignments," *Bioinformatics,* vol. 18, pp. 306-314, 2002.

[9] G.M. Cooper, E.A. Stone, G. Asimenos, NISC Comparative Sequencing Program, E.D. Green, S. Batzoglou, and A. Sidow, "Distribution and Intensity of Constraint in Mammalian Genomic Sequence," *Genome Research,* vol. 15, pp. 901-913, 2005.

[10] R.C. Edgar, "MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput," *Nucleic Acids Research,* vol. 32, no. 5, pp. 1792-1797, 2004.

[11] M. Errami, C. Geourjon, and G. Deléage, "Detection of Unrelated Proteins in Sequences Multiple Alignments by Using Predicted Secondary Structures," *Bioinformatics,* vol. 19, no. 4, pp. 506-512, 2003.

[12] J. Felsenstein, "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach," *J. Molecular Evolution,* vol. 17, pp. 368-376, 1981.

[13] N. Kaplan, O. Sasson, U. Inbar, M. Friedlich, M. Fromer, H. Fleischer, E. Portugaly, N. Linial, and M. Linial, "ProtoNet 4.0: A Hierarchical Classification of One Million Protein Sequences," *Nucleic Acids Research,* vol. 33, pp. D216-D218, 2005.

[14] S. Karlin and S.F. Altschul, "Methods for Assessing the Statistical Significance of Molecular Sequence Features by Using General Scoring Schemes," *Proc. Nat'l Academy of Science of the USA,* vol. 87, no. 6, pp. 2264-2268, Mar. 1990.

[15] S. Karlin and S.F. Altschul, "Applications and Statistics for Multiple High-Scoring Segments in Molecular Sequences," *Proc. Nat'l Academy of Science of the USA,* vol. 90, pp. 5873-5877, June 1993.

[16] W. Kent, C.W. Sugnet, T.S. Furey, K. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler, "The Human Genome Browser at UCSC," *Genome Research,* vol. 12, no. 6, pp. 996-1006, 2002.

[17] S. Kumar and A. Filipski, "Multiple Sequence Alignment: In Pursuit of Homologous DNA Positions," *Genome Research,* vol. 17, no. 2, pp. 127-135, Feb. 2007.

[18] T. Lassmann and E.L.L. Sonnhammer, "Automatic Assessment of Alignment Quality," *Nucleic Acids Research,* vol. 33, no. 22, pp. 7120-7128, 2005.

[19] E. Margulies, M. Blanchette, NISC Comparative Sequencing Program, D. Haussler, and E. Green, "Identification and Characterization of Multi-Species Conserved Sequences," *Genome Research,* vol. 13, no. 12, pp. 2507-2518, 2003.

[20] E.H. Margulies, G.M. Cooper, G. Asimenos, D.J. Thomas, C.N. Dewey, A. Siepel, E. Birney, D. Keefe, A.S. Schwartz, M. Hou, J. Taylor, S. Nikolaev, J.I. Montoya-Burgos, A. Lvytynoja, S. Whelan, F. Pardi, T. Massingham, J.B. Brown, P. Bickel, I. Holmes, J.C. Mullikin, A. Ureta-Vidal, B. Paten, E.A. Stone, K.R. Rosenbloom, W.J. Kent, G.G. Bouffard, X. Guan, N.F. Hansen, J.R. Idol, V.V. Maduro, B. Maskeri, J.C. McDowell, M. Park, P.J. Thomas, A.C. Young, R.W. Blakesley, D.M. Muzny, E. Sodergren, D.A. Wheeler, K.C. Worley, H. Jiang, G.M. Weinstock, R.A. Gibbs, T. Graves, R. Fulton, E.R. Mardis, R.K. Wilson, M. Clamp, J. Cuff, S. Gnerre, D.B. Jaffe, J.L. Chang, K. Lindblad-Toh, E.S. Lander, A. Hinrichs, H. Trumbower, H. Clawson, A. Zweig, R.M. Kuhn, G. Barber, R. Harte, D. Karolchik, M.A. Field, R.A. Moore, C.A. Matthewson, J.E. Schein, M.A. Marra, S.E. Antonarakis, S. Batzoglou, N. Goldman, R. Hardison, D. Haussler, W. Miller, L. Pachter, E.D. Green, and A. Sidow, "Analyses of Deep Mammalian Sequence Alignments and Constraint Predictions for 1% of the Human Genome," *Genome Research,* vol. 17, no. 6, pp. 760-774, June 2007.

[21] W. Miller, "Comparison of Genomic Sequences: Solved and Unsolved Problems," *Bioinformatics,* vol. 17, pp. 391-397, 2000.

[22] S.B. Needleman and C.D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *J. Molecular Biology,* vol. 48, pp. 443-453, 1970.

[23] J. Pei and N.V. Grishin, "AL2CO: Calculation of Positional Conservation in a Protein Sequence Alignment," *Bioinformatics,* vol. 17, pp. 700-712, 2001.

[24] P. Pevzner, *Computational Molecular Biology: An Algorithmic Approach,* chapter 6, pp. 102-109. MIT Press, 2000.

[25] D.A. Pollard, C.M. Bergman, J. Stoye, S.E. Celniker, and M.B. Eisen, "Benchmarking Tools for the Alignment of Functional Noncoding DNA," *BMC Bioinformatics,* vol. 5, article 6, 2004.

[26] D.A. Pollard, A.M. Moses, V.N. Iyer, and M.B. Eisen, "Detecting the Limits of Regulatory Element Conservation and Divergence Estimation Using Pairwise and Multiple Alignments," *BMC Bioinformatics,* vol. 7, article 376, 2006.

[27] S. Prabhakar, F. Poulin, M. Shoukry, V. Afzal, E.M. Rubin, O. Couronne, and L.A. Pennacchio, "Close Sequence Comparisons Are Sufficient to Identify Human CIS-Regulatory Elements," *Genome Research,* vol. 16, no. 7, pp. 855-863, July 2006.

[28] A. Prakash and M. Tompa, "Discovery of Regulatory Elements in Vertebrates through Comparative Genomics," *Nature Biotechnology*, vol. 23, no. 10, pp. 1249-1256, 2005.
[29] A. Prakash and M. Tompa, "Statistics of Local Multiple Alignments," *Bioinformatics*, vol. 21, pp. i344-i350, 2005.
[30] A. Prakash and M. Tompa, "Measuring the Accuracy of Genome-Size Multiple Alignments," *Genome Biology*, vol. 8, no. 6, p. R124, 2007.
[31] M.S. Rosenberg, "Multiple Sequence Alignment Accuracy and Evolutionary Distance Estimation," *BMC Bioinformatics*, vol. 6, article 278, 2005.
[32] W.L. Ruzzo and M. Tompa, "A Linear Time Algorithm for Finding All Maximal Scoring Subsequences," *Proc. Seventh Int'l Conf. Intelligent Systems for Molecular Biology*, pp. 234-241, Aug. 1999.
[33] N. Saitou and M. Nei, "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Molecular Biology and Evolution*, vol. 4, pp. 406-425, 1987.
[34] A.A. Schäffer, L. Aravind, T.L. Madden, S. Shavirin, J.L. Spouge, Y.I. Wolf, E.V. Koonin, and S.F. Altschul, "Improving the Accuracy of PSI-BLAST Protein Database Searches with Composition-Based Statistics and Other Refinements," *Nucleic Acids Research*, vol. 29, no. 14, pp. 2994-3005, 2001.
[35] A.A. Schäffer, Y.I. Wolf, C.P. Ponting, E.V. Koonin, L. Aravind, and S.F. Altschul, "IMPALA: Matching a Protein Sequence against a Collection of PSI-BLAST-Constructed Position-Specific Score Matrices," *Bioinformatics*, vol. 15, no. 2, pp. 1000-1011, 1999.
[36] A. Siepel, G. Bejerano, J. Pedersen, A. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L. Hillier, S. Richards, G. Weinstock, R.K. Wilson, R. Gibbs, W. Kent, W. Miller, and D. Haussler, "Evolutionarily Conserved Elements in Vertebrate, Insect, Worm, and Yeast Genomes," *Genome Research*, vol. 15, no. 8, pp. 1034-1050, 2005.
[37] A.F.A. Smit, R. Hubley, and P. Green Repeatmasker Open-3.0, http://www.repeatmasker.org, 2004.
[38] P.H.A. Sneath and R.R. Snokal, *Numerical Taxonomy*, pp. 230-234. W.H. Freeman, 1973.
[39] R.L. Tatusov and D.J. Lipman, unpublished, http://blast.wustl.edu/pub/dust, 2007.
[40] "The ENCODE (ENCyclopedia Of DNA Elements) Project," *Science*, vol. 306, no. 5696, pp. 636-640, The ENCODE Project Consortium, 2004.
[41] J. Thompson, F. Plewniak, and O. Poch, "BAliBASE: A Benchmark Alignments Database for the Evaluation of Multiple Sequence Alignment Programs," *Bioinformatics*, vol. 15, no. 1, pp. 87-88, 1999.
[42] J.D. Thompson, F. Plewniak, and O. Poch, "A Comprehensive Comparison of Multiple Sequence Alignment Programs," *Nucleic Acids Research*, vol. 27, pp. 2682-2690, 1999.
[43] J.D. Thompson, F. Plewniak, R. Ripp, J.C. Thierry, and O. Poch, "Towards a Reliable Objective Function for Multiple Sequence Alignments," *J. Molecular Biology*, vol. 314, pp. 937-951, 2001.
[44] J.D. Thompson, V. Prigent, and O. Poch, "LEON: Multiple Alignment Evaluation of Neighbours," *Nucleic Acids Research*, vol. 32, no. 4, pp. 1298-1307, 2004.
[45] A.X. Wang, W.L. Ruzzo, and M. Tompa, "How Accurately Is ncRNA Aligned Within Whole-Genome Multiple Alignments?" *BMC Bioinformatics*, vol. 8, article 417, 2007.
[46] L. Wang and T. Jiang, "On the Complexity of Multiple Sequence Alignment," *J. Computational Biology*, vol. 1, pp. 337-348, 1994.
[47] D. Wheeler, T. Barrett, D. Benson, S. Bryant, K. Canese, D. Church, M. DiCuccio, R. Edgar, S. Federhen, W. Helmberg, D. Kenton, O. Khovayko, D. Lipman, T. Madden, D. Maglott, J. Ostell, J. Pontius, K. Pruitt, G. Schuler, L. Schriml, E. Sequeira, S. Sherry, K. Sirotkin, G. Starchenko, T. Suzek, R. Tatusov, T. Tatusova, L. Wagner, and E. Yaschenko, "Database Resources of the National Center for Biotechnology Information," *Nucleic Acids Research*, vol. 33, pp. D39-D45, 2005.

Amol Prakash received the PhD from the Department of Computer Science and Engineering, University of Washington, in 2006. He is a senior research scientist in the Biomarker Research Initiative in Mass Spectrometry (BRIMS) Center, Thermo Fisher Scientific, Cambridge, Massachusetts, one of the largest manufacturers of scientific and analytical instruments. In this position, he develops innovative mass spectrometry-based workflows that can help scientists around the world in their biomarker-based projects. His research interests include mass spectrometry-based proteomics and comparative genomics.

Martin Tompa received the PhD degree in computer science from the University of Toronto in 1978. For the next seven years, he was on the computer science faculty at the University of Washington, Seattle. From 1985 to 1989, he worked at IBM Research in the T.J. Watson Research Center and became the manager of its Theory of Computation Group. In 1989, he rejoined the faculty at the University of Washington, where he is currently a professor of computer science and engineering and an adjunct professor of genome sciences. His research interests include biological sequence analysis, regulatory analysis, and comparative genomics.