TreeRefiner: A Tool for Refining a Multiple Alignment on a Phylogenetic Tree

Aswath Manohar and Serafim Batzoglou

Department of Computer Science
Stanford University, Stanford CA 94305
amanohar@cs.stanford.edu, serafim@cs.stanford.edu

Abstract

We present TreeRefiner, a tool for refining multiple alignments of biological sequences. Given a multiple alignment, a phylogenetic tree, and scoring parameters as input, TreeRefiner optimizes the sum-of-pairs function in a restricted three-dimensional space around the alignment. At each internal node of the unrooted tree, the multiple alignment is projected to the sub-alignments corresponding to the three neighboring nodes, and three-dimensional dynamic programming is performed within a user-specified radius r around the original alignment. We test TreeRefiner on simulated sequences aligned by several popular tools, and demonstrate substantial improvements in the percentage of correctly aligned positions.

1. Introduction

Multiple alignment is one of the main steps in the analysis of biological sequences. Comparative methods, which use sequence conservation as a signal to identify regions of functional importance, have been applied to gene identification [5, 7, 1, 25, 43], regulatory motif finding [27, 24, 26, 42], noncoding RNA gene finding and structural determination [33], phylogenetic and evolutionary analysis [13, 35], identification of conserved domains and residues [4, 18], and characterization of protein families [36].

Because multiple alignment is an intractable computational problem [41], practical tools rely on heuristic methodologies such as progressive alignment [6, 38, 21, 40, 19] in which the multiple alignment is constructed by progressive application of a two-dimensional alignment procedure between two sequences or intermediate alignments, profile-based alignment [16], or greedy assemblage of segment-to-segment comparisons [30]. Using these methods, protein aligners such as CLUSTALW [40], TCOFFEE [31], MUSCLE [17] and PROBCONS [15] have been successful in providing reasonable multiple alignments of large protein families, and genomic aligners such as MLAGAN [11], MAVID [8], and TBA [10] can handle whole mammalian genomes when combined with pre-existing genomic maps or with automated genomic alignment pipelines [34, 14, 12].

Heuristic procedures usually sacrifice accuracy in favor of computational efficiency. During progressive alignment, for instance, errors during early steps are propagated to the final output. *Iterative refinement* is a methodology for improving an existing multiple alignment by iterative application of a refinement procedure: during each refinement, a sequence is removed and realigned to the remaining multiple alignment [20, 9]. This procedure, which is guaranteed to maintain or improve the alignment score, is applied repeatedly with every sequence until the score converges. Variants of the method have been introduced for efficiency, such as requiring the new alignment not to deviate by more than a fixed distance from the original alignment [3].

Existing iterative refinement methods rely on realignment steps between *pairs* of sequences or partial alignments. In this paper, we introduce a tool based on three-dimensional alignment. Alignment between three (multi-)sequences is a natural step: usually, related biological sequences are connected into binary phylogenetic trees, whose internal nodes always have degree 3. Therefore, a promising way to align sequences is to integrate information at each internal node of the tree, taking into account the two daughter nodes and the outgroup node simultaneously. Progressive alignment in contrast, first fixes an alignment between the two daughter nodes, and then merges that alignment with the third

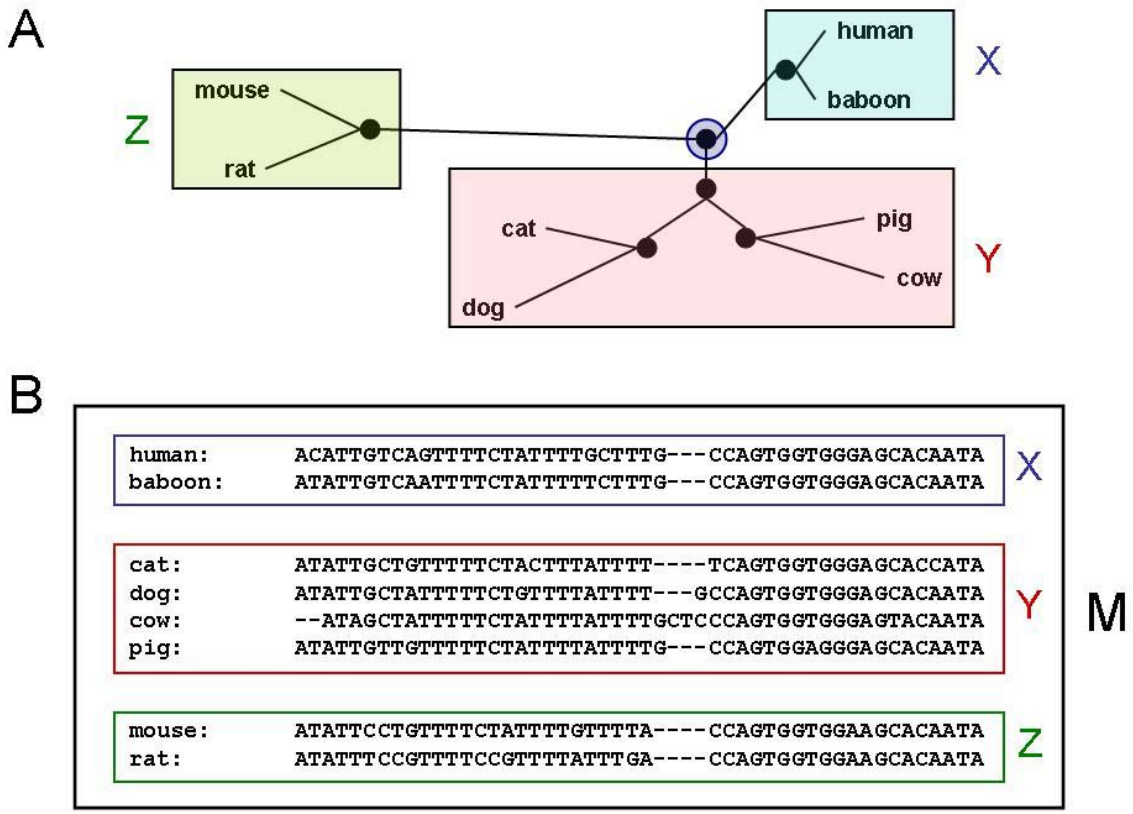

Figure 1. Projection of a multiple alignment *M* into the three daughter alignments *X*, *Y*, and *Z*, with respect to an internal node of the phylogenetic tree. **A.** The marked internal (non-leaf) node divides the species' tree into three subtrees. **B.** The subtrees correspond to projections of *M* into three subalignments *X*, *Y*, and *Z*. Note that in order to obtain a subalignment, gaps in M that are common to all sequences within the subset have to be removed.

node—losing the opportunity to correct errors made during the first step.

We tested the ability of TreeRefiner to improve multiple alignments produced by leading multiple alignment tools, on sets of nucleotide sequences that were generated by the Rose program [37] that simulates sequence evolution on a tree. Based on standard measures of alignment accuracy in the case where the "true" alignment is known, we find that TreeRefiner improves the accuracy of most aligners significantly.

2. Algorithms

TreeRefiner takes the following four inputs: (1) a multiple alignment between *K* sequences; (2) an unrooted phylogenetic tree connecting the sequences; (3) scoring parameters, including a gap-open penalty *d*, a gap-extension penalty *e*, and a substitution matrix for nucleotides or amino acids; and (4) a radius *r* that limits the search space on which three-dimensional dynamic programming will be performed.

At each internal node of the phylogenetic tree, which is traversed bottom-up, TreeRefiner applies the following basic refinement procedure:

1. The current multiple alignment *M* is projected into the sub-alignments *X*, *Y*, and *Z*, representing the three daughter nodes (Figure 1). *M* defines a path of points $\{(x_1, y_1, z_1),\ldots, (x_m, y_m, z_m)\}$, where *M* contains *m* aligned columns, each x_i is a column of the projected alignment *X* and $x_1 \le \ldots \le x_m$, and similarly for each y_i and z_i (Figure 2a).
2. A limited volume *R* on the three-dimensional space *X*×*Y*×*Z* is defined by $R = \{(x, y, z) \mid \exists\ (x', y', z') \in M$ s.t. $(x' - r \le x \le x'+r)$ & $(y' - r \le x \le y' + r)$ & $(z' - r \le x \le$

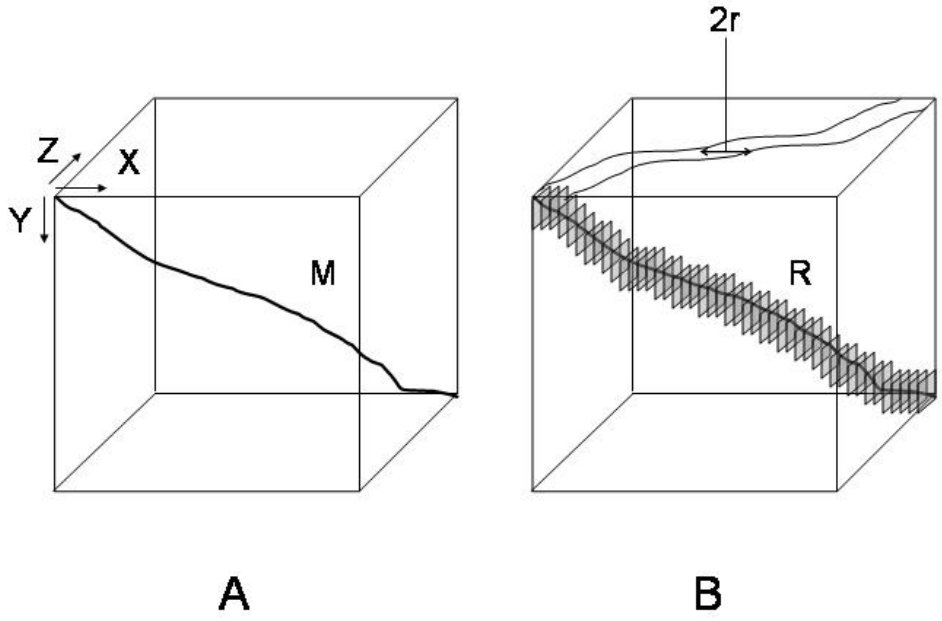

Figure 2. Limited area around the input alignment where refinement takes place. **A.** Given an initial alignment M, and three projections X, Y, and Z, M defines a path of points $\{(x_1, y_1, z_1),\ldots, (x_m, y_m, z_m)\}$. **B.** A limited area R is defined to be all points that fall within a Manhattan distance r from the original alignment. The boundaries of R's projection on the X-Z plane are shown.

$z' + r$)}. That is, the limited area R contains all points that fall within a Manhattan distance r from the original alignment (Figure 2b). It follows that R contains $O(r^2m)$ cells.

3. Dynamic programming is performed on the Cartesian product $R \times S$ of volume R and set of column match/gap states S. The set S contains 7 states: one that emits three positions (x, y, z) from the three sub-alignments, and 6 states corresponding to all possible gap patterns: { (x, y, –), (x, –, z), (–, y, z), (x, –, –), (–, y, –), (–, –, z) }. Each state is implemented as a dynamic programming matrix, to optimize the sum-of-pairs score between all original K sequences, under the user-specified scoring parameters.

Table 1. *The symmetric pessimistic gap scoring function for estimating and penalizing the number of gap events.* In each axis we show the four possibilities of consecutive characters (gap vs. non-gap) in a given sequence. In the table we show how many gaps are counted locally by (symmetric pessimistic, pessimistic, optimistic) where the latter two are given in (Kececioglu and Zhang 1998).

	--	-x	x-	xx
--	0, 0, 0	½, 1, 0	½, 0, 0	0, 0, 0
-x		0, 0, 0	1, 1, 1	½, 0, 0
x-			0, 0, 0	½, 1, 1
xx				0, 0, 0

The *sum-of-pairs* score is defined as follows. In each column, substitution scores are added for each pair of nucleotides. In addition each gap is labeled as 'O' (gap open) or 'G' (gap extension), and each nucleotide is labeled as 'C' (gap-closing nucleotide) or 'N' (other nucleotide), Gaps are labeled 'O' when preceded by a nucleotide in the same sequence, and 'G' otherwise. Nucleotides are labeled 'C' when preceded by a gap in the same sequence, and 'N' otherwise. The column then receives a gap penalty defined by $(d/2) \times (\#O) \times (\#N + \#C + G) + e \times (\#G) \times (\#N + \#C) + (d/2) \times (\#C) \times (\#N + \#O + \#G)$. By attributing ½ the

Table 2. *Performance of TreeRefiner on 15 kb-long alignments produced by CLUSTALW and DIALIGN.* Both tools were run with default parameters. For each of the four parameter sets, the best SP or TC number before refinement is shown in bold; the best SP or TC number overall (including before and after refinement) is shown in bold and underlined. We note that while both tools benefit from refinement, DIALIGN benefits significantly more. One possible explanation is that DIALIGN produces alignments that are at least as good as those of CLUSTALW on the larger scale mapping of the sequences, but suffer on very local nucleotide-level accuracy.

Alignment	CLUSTALW			DIALIGN		
	TR radius	SP	TC	TR radius	SP	TC
((human baboon) (mouse rat))	none	**0.957**	**0.883**	none	0.902	0.788
	1	0.960	0.887	1	0.938	0.847
	2	0.961	0.891	2	0.954	0.876
	4	0.965	0.899	4	0.964	0.896
	8	**0.965**	**0.899**	8	**0.965**	0.897
((human baboon) (mouse rat)) + higher mutation rate	none	**0.940**	**0.838**	none	0.838	0.719
	1	0.942	0.840	1	0.900	0.801
	2	0.944	0.847	2	0.903	0.844
	4	0.945	0.851	4	0.927	0.873
	8	**0.946**	0.854	8	0.934	**0.879**
(((pig cow) (dog cat)) ((human baboon) (mouse rat)))	none	**0.888**	**0.649**	none	0.816	0.538
	1	0.902	0.673	1	0.881	0.653
	2	0.909	0.693	2	0.903	0.701
	4	0.918	0.716	4	0.927	0.750
	8	0.921	0.722	8	**0.934**	**0.766**
(((pig cow) (dog cat)) ((human baboon) (mouse rat))) + higher mutation rate	none	**0.793**	**0.446**	none	0.732	0.422
	1	0.813	0.475	1	0.799	0.509
	2	0.823	0.491	2	0.835	0.571
	4	0.832	0.515	4	0.874	0.640
	8	0.835	0.525	8	**0.882**	**0.654**

gap-open penalty in the beginning of a gap, and ½ in the end, the function is symmetric with respect to initiation of dynamic programming in the top-left-front corner or the bottom-right-back corner of the search space, and partially avoids artificial effects that sum-of-pairs exhibits when combined with affine gaps.

Note that gaps are not penalized according to the transitions between states in S, but rather according to the gap patterns of the individual sequences. As an example, consider scoring the state transition $(x_5, y_3, z_8) \rightarrow (x_6, -, z_9)$, where each x_i, y_j, z_k is a column of an alignment X, Y, and Z, as follows: x_5 = (A, A); x_6 = (C, C); y_3 = (A, A); z_8 = (A, G); z_9 = (C, –). Then, the transition is (A, A, A, A, A, –) → (C, C, –, –, C, –). The second column receives 3 C-C matches. Additionally, its gap pattern is labeled as (N, N, O, O, N, G) and receives gap penalty of $(d/2) \times 2 \times (3 + 0 + 1) + e \times 1 \times (3 + 0) + (d/2) \times 0 \times (3 + 2 + 1) = 4d + 3e$.

Aligning two multiple alignments under the affine gap sum-of-pairs scoring function is NP-hard [28, 22], and therefore aligning three alignments, as in our case, is obviously intractable. Several heuristic scoring functions for penalizing gaps have been defined, such as the *quasi-natural* function [2], the *optimistic* and the *pessimistic* gap functions [23]. Our scoring function and optimization procedure is also necessarily heuristic, in that the number of gap events in the projected pairs of sequences cannot be estimated exactly. In Table 1 we compare the number of gap-open penalties incurred according to our function to those estimated by pessimistic and optimistic gap counts. Our method can be considered as *symmetric pessimistic*, in that it penalizes gap openings symmetrically with gap closings, while it overestimates the number of gap events in several situations.

3. Results

To test the performance of TreeRefiner, reference sequence alignments are created by using a probabilistic evolution simulator based on Rose.

Table 3. *Performance of TreeRefiner on 100 kb-long alignments produced by MLAGAN, MAVID, and TBA.* All tools were run with default parameters. For each of the four parameter sets, the best SP or TC number before refinement is shown in bold; the best SP or TC number overall (including before and after refinement) is shown in bold and underlined. TBA did not return an alignment in the high-divergence 8-species input. These results indicate that MLAGAN and MAVID alignments benefit considerably from TreeRefiner, while TBA is significantly more accurate to start with, and does not benefit from refinement.

Alignment	MLAGAN			MAVID			TBA		
	TR radius	SP	TC	TR radius	SP	TC	TR radius	SP	TC
((human baboon) (mouse rat))	none	0.927	0.847	none	0.941	0.869	none	**0.961**	**0.900**
	1	0.956	0.894	1	0.953	0.890	1	0.961	0.900
	2	0.962	0.906	2	0.958	0.899	2	0.961	0.900
	4	0.965	0.911	4	0.961	0.903	4	0.961	0.898
	8	**0.965**	**0.912**	8	0.963	0.905	8	0.961	0.898
((human baboon) (mouse rat)) + higher mutation rate	none	0.913	0.823	none	0.849	0.745	none	**0.952**	**0.890**
	1	0.939	0.866	1	0.860	0.763	1	0.949	0.883
	2	0.944	0.875	2	0.866	0.772	2	0.947	0.881
	4	0.948	0.883	4	0.872	0.781	4	0.947	0.880
	8	0.948	0.884	8	0.879	0.791	8	0.947	0.880
(((pig cow) (dog cat)) ((human baboon) (mouse rat)))	none	0.868	0.684	none	0.848	0.628	none	**0.924**	**0.769**
	1	0.914	0.742	1	0.878	0.669	1	0.927	0.762
	2	0.926	0.757	2	0.892	0.693	2	0.928	0.760
	4	0.932	0.767	4	0.901	0.708	4	0.929	0.757
	8	**0.934**	**0.769**	8	0.907	0.717	8	0.929	0.757
(((pig cow) (dog cat)) ((human baboon) (mouse rat))) + higher mutation rate	none	**0.792**	**0.547**	none	0.660	0.359	none	*	*
	1	0.864	0.641	1	0.682	0.385	1	*	*
	2	0.884	0.669	2	0.694	0.398	2	*	*
	4	0.896	0.687	4	0.706	0.414	4	*	*
	8	**0.901**	**0.695**	8	0.716	0.423	8	*	*

Guided by an evolutionary tree, a family of related sequences is created starting with a random ancestral sequence under the HKY model with transition/transversion bias, substitution rate, and insertion/deletion rates set to mimic the sequence divergence properties of real sequences as estimated [13]. During this artificial evolutionary process, the `correct' multiple sequence alignment is created, and can be used as reference for measuring alignment accuracy. Similar benchmarking strategies based on simulations have been applied previously [32, 10].

We tested improvement conferred by TreeRefiner with respect to alignments generated by CLUSTALW, DIALIGN, MAVID, MLAGAN, and TBA. First, the unaligned derived sequences generated by Rose were aligned with the tested aligners. Then the output multiple alignments were fed to TreeRefiner to obtain a refined multiple alignment. To compare the accuracy of the original and improved alignments, two scores SP (sum of pairs) and TC (total columns) were used. SP is the ratio of the number of correctly aligned pairs of letters in the test alignment to the number of aligned pairs in the reference alignment. TC is the ratio of the number of correctly aligned columns in the test alignment to the number of columns in the reference alignment. Both SP and TC range from 1.0 for perfect agreement to 0.0 for no agreement.

In the case of CLUSTALW and DIALIGN, for efficiency reasons we generated simulated sequences of length 15 kb. For MLAGAN, MAVID, and TBA, which scale well with sequence length, we instead generated simulated sequences of length 100 kb. We run Rose on two trees, each with two sets of parameters. We marked the first tree as ((human baboon) (mouse rat)), and the second tree as (((pig cow) (dog cat)) ((human baboon) (mouse rat))). On each tree, we chose the first parameter set to mimic sequence divergence between the corresponding species, as measured [13] on a region harboring the

Cystic Fibrosis Transmembrane Conductance Regulator gene [39]. The second parameter set kept gap parameters the same, but increased the mutation rates 40% from their previous values. All simulations and parameters are described in detail in our website, http://treerefiner.stanford.edu.

Tables 2 and 3 show the performance of TreeRefiner on alignments produced by the different programs. As seen in the tables, TreeRefiner improves the accuracy of most programs under all four different parameter sets, both in terms of the SP and the TC measure. As we increase the radius of dynamic programming area where TreeRefiner is run, we see improvements in accuracy, and note that in most cases we experience diminishing returns as the radius increases from 4 to 8.

We note that some of the alignment tools benefit much more than other tools by application of TreeRefiner on their output. For example, DIALIGN seems to improve considerably more than CLUSTALW: while the original CLUSTALW alignments are always more accurate than the ones by DIALIGN according to the SP and TC measures, the DIALIGN alignments tend to become more accurate as soon as TreeRefiner is applied with radius at least 4. MLAGAN and MAVID also improve considerably with refinement. The only tool that does not benefit from refinement is TBA, which performs extremely well in these simulations and refinement results in similar or slightly lower accuracy, depending on the example. TBA is known to be the best aligner in simulation benchmarks [10].

We observed that the running time of TreeRefiner, as expected, grows approximately linearly with length of the input multiple alignment, linearly with the number of species, and exponentially with radius. In the cases of CLUSTALW and DIALIGN alignments, TreeRefiner had significantly lower running time (at least 10 times lower for CLUSTALW, and 3 times lower for DIALIGN, for radius 8 on the 8-species alignment). Compared to the other aligners, TreeRefiner was significantly slower with high radius. For example, a radius of 8 takes approximately 800 seconds on a multiple alignment of 8 sequences of length 100,000 each, on a 3.2 GHz Pentium IV machine. However, running time is practical for long alignments: we refined the MLAGAN alignment of the CFTR region in 9 species (4.3 million columns) in 24 minutes with radius 2. Although the "correct" alignment is not known as in simulations, and therefore we cannot measure local nucleotide-level improvements rigorously, the resulting alignment looks significantly improved by visual inspection. We include examples of regions before and after improvement, in Table 4.

Supplemental results are available in our webpage, http://treerefiner.stanford.edu. We include the source code and executable for TreeRefiner, with documentation of how to run it. We also include our test scripts, test input sequences, as well as resulting alignments of running each tool before and after refinement. Finally, we include the alignment of the CFTR region before and after refinement, as well as automatically generated snapshots of regions that are significantly improved according to a score difference threshold.

4. Discussion

TreeRefiner is a practical tool for improving the affine sum-of-pairs score of a multiple alignment on a limited search area within the original alignment. TreeRefiner is available as public domain software, at http://treerefiner.stanford.edu. The symmetric-pessimistic scoring function used in this paper is a version of sum-of-pairs with affine gaps, with user-defined parameters. Users can experiment with alternative scoring models by modifying the per-column scoring function in the source code, for which documentation is available.

The refinement algorithm used by TreeRefiner can make only local adjustments to an alignment: with radius r, a segment of arbitrary length can move at most r positions, or a segment of length r can move an arbitrary number of positions, with respect to the input alignment. No long-range improvements can be made, such as aligning correctly a protein-coding exon that was previously missed. Local improvements are important in applications that require high accuracy at the level of single letters or short features. Examples include phylogenetic analysis and estimates of evolutionary rates, where the number of substitutions should not be overestimated, comparative gene recognition, where features such as the 'ATG', splice sites, and stop codons should be aligned exactly, regulatory element finding, where the motifs of interest are short and often difficult to align, and noncoding RNA structure prediction and detection where, again, features such as stems and bulges are short.

Development of rigorous methods for measuring the accuracy of real genomic DNA alignments is an open problem [29]. In lieu of that, simulations are a reasonable alternative [32, 10]. While we cannot quantify rigorously the improvement in accuracy that TreeRefiner will produce on real sequences, our simulations indicate that improvements are significant with respect to alignments produced by several popular aligners that we tested.

Table 4. *Examples of TreeRefiner improvements on the MLAGAN alignment of the CFTR region.* In both cases, changes are subtle and involve insertion or shifting of gaps in several sequences. Visual inspection points to several areas of the alignment such as the two shown here, where the large-scale map remains the same, but the nucleotide-level alignment is significantly improved after refinement. Quantifying the nucleotide-level accuracy of alignments of real genomic DNA alignments remains an open problem.

Before Refinement. Score: 103967; Column Number: 697953.

```
Baboon : GAGCCCAGTGCTTTGAGAATG-TCAATGCAAAATTATAATAATTACTTATC
Chimp  : GAGCCCAGTGCTTTGAGAATG-CCAATGCAAAATTATAATAATTACTTATT
Human  : GAGCCCAGTGCTTTGAGAATG-CCAATGCAAAATTATAATAATTACTTATT
Cat    : GAACTCAGTGCTTTGAGACTG-TTAATGCAAAATTATGACAACT--TTATT
Dog    : GAGCTCAGTACTTTGAGAATG-TCAATGCAAAATTATAATAATTGCTCATT
Cow    : GAGGCCAGTGCTTTGAGAATG-CCAATGCAAAATTATAATAATTGCTTATT
Pig    : GATCCCAGTGCTTTGAGAATG-CCAATGCAAAATT---ATAATTGCTTCTT
Mouse  : GAGTCCAATACTTAAGAGAATGTCAATACAAAATTAAAATAATTGGTCATT
Rat    : GAGCCCAATACTTTGAGATTGTCAATACAAAATTAAAAATAATTGCTCATT
```

After Refinement. Score: 122382; Column Number: 704386.

```
Baboon : GAGCCCAGTGCTTT-GAGAATGTCAATGCAAAATTATAA-TAATTACTTATC
Chimp  : GAGCCCAGTGCTTT-GAGAATGCCAATGCAAAATTATAA-TAATTACTTATT
Human  : GAGCCCAGTGCTTT-GAGAATGCCAATGCAAAATTATAA-TAATTACTTATT
Cat    : GAACTCAGTGCTTT-GAGACTGTTAATGCAAAATTATGA-CAACT--TTATT
Dog    : GAGCTCAGTACTTT-GAGAATGTCAATGCAAAATTATAA-TAATTGCTCATT
Cow    : GAGGCCAGTGCTTT-GAGAATGCCAATGCAAAATTATAA-TAATTGCTTATT
Pig    : GATCCCAGTGCTTT-GAGAATGCCAATGCAAAATTA----TAATTGCTTCTT
Mouse  : GAGTCCAATACTTAAGAGAATGTCAATACAAAATTAAAA-TAATTGGTCATT
Rat    : GAGCCCAATACTTT-GAGATTGTCAATACAAAATTAAAAATAATTGCTCATT
```

Before Refinement. Score: 103967; Column Number: 697953.

```
Baboon : CATGCTAAGACCCATTTTAGCTCTGATTTTCTGTGAGTCATAGCAGAGGA
Chimp  : CATGCTAAGACTCATTTTAGCTCTGATTTTCTGTGAGTCATAGCAGAGGG
Human  : CATGCTAAGACTCATTTTAGCTCTGATTTTCTGTGAGTCATAGCAGAGGG
Cat    : ATCTCTAAT-CCATTTTAGCTCGATTTTTTTGTGTGTGTCATAGCAGGGG
Dog    : ATCTCCAATATTCATTTTACATCTGATTTTTTGGTGTATTGTAGCAGGGG
Cow    : GTCTCGAAGGTTCATTTTCACCCCAGTTTTCT-GTGAGTCATGGCAGGGA
Pig    : ATTTTTGAGGGTCATTTTCCTCCAGTTTTCTG-TGAGTCAATGGCATGGT
Mouse  : GTCCTGCAGATTCCTTTTTGCTTTGAATGTCTGTGAGTCACGTTACAAGG
Rat    : GTCTTGCAGATTCATTTTTGCTTTGAATTTCTGTGAGTCACGTCACAAGG
```

After Refinement. Score: 122382; Column Number: 704386.

```
Baboon : CATG----CTAAGACCCATTTTAGCTCTGATTTTCTG--TGAGTCATAGCAGAGGA
Chimp  : CATG----CTAAGACTCATTTTAGCTCTGATTTTCTG--TGAGTCATAGCAGAGGG
Human  : CATG----CTAAGACTCATTTTAGCTCTGATTTTCTG--TGAGTCATAGCAGAGGG
Cat    : GATGATCTCTAA--TCCATTTTAGCTCGATTTTTTTGTGTGTGTCATAGCAGGGGG
Dog    : GATGATCTCCAATATTCATTTTACATCTGATTTTTTG-GTGTATTGTAGCAGGGGG
Cow    : GAGGGTCTCGAAGGTTCATTTTCACCCCAGTTTTCTG--TGAGTCATGGCAGGGAG
Pig    : GATGATTTTTGAGGGTCATTTTC-CTCCAGTTTTCTG--TGAGTCAATGGCATGGT
Mouse  : AGTGGTCCTGCAGATTCCTTTTTGCTTTGAATGTCTG--TGAGTCACGTTACAAGG
Rat    : AGCAGTCTTGCAGATTCATTTTTGCTTTGAATTTCTG--TGAGTCACGTCACAAGG
```

Acknowledgements

We thank Chuong B. Do for help with development of TreeRefiner, and Mike Brudno for valuable discussions on the concepts presented here. This work was supported in part by NSF grant 0312459 and by the NSF CAREER Award.

References

1. Alexandersson M, Cawley S, Pachter L. SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Research* 13:496-502, 2003.
2. Altschul SF. Gap costs for multiple sequence alignment. Journal of Theoretical Biology 138: 297-309, 1989.
3. Anson EL, Myers EW. Re-Aligner: A program for refining DNA sequence multi-alignments. *Journal of Computational Biology* **4:**369–383, 1997.
4. Attwood TK. The PRINTS database: a resource for identification of protein families. *Briefings in Bioinformatics* 3(3):252-263, 2002.
5. Bafna V, Huson DH. The Conserved Exon Method for gene finding. I*SMB-00: Proceedings of the Eight International Conference on Intelligent systems for Molecular Biology*. **8:** 3-12, 2000.
6. Barton G.J. and M.J.E. Sternberg. A strategy for the rapid multiple alignment of protein sequences. *Journal of Molecular Biology* 198:327–337, 1987.
7. Batzoglou S, Pachter L, Mesirov J, Berger B, Lander E. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research* 10:950-958, 2000.
8. Bray N, Pachter L. MAVID: Constrained ancestral alignment of multiple sequences. *Genome Research* **13:**693–699, 2004.
9. Berger MP, Munson PJ. A novel randomized iterative strategy for aligning multiple protein sequences. *CABIOS* 7:479-484, 1991.
10. Blanchette M, Kent JW, Riemer C, Elnitski L, Smit AF, Roskin KM, Baertsch R, Rosenbloom K, Clawson H, Green ED, et al. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research* **14:**708-715, 2004.
11. Brudno M, Do CB, Cooper GM, Kim MF, Davydov E, NISC Comparative Sequencing Program, Green ED, Sidow A, Batzoglou S. LAGAN and Multi-LAGAN: efficient tools for large scale multiple alignment of genomic DNA. *Genome Research* 13:721-731, 2003.
12. Brudno M, Poliakov A, Salamov A, Cooper GM, Sidow A, Rubin EM, Solovyev V, Batzoglou S, Dubchak I. Automated Whole-Genome Multiple Alignment of Rat, Mouse, and Human. *Genome Research* **14:**685–692, 2004.
13. Cooper GM, Brudno M, NISC Comparative Sequencing Program, Green ED, Batzoglou S, Sidow A. Quantitative estimates of sequence divergence for comparative analysis of mammalian genomes. *Genome Research* 13:813-820, 2003.
14. Couronne O, Poliakov A, Bray N, Ishkhanov T, Ryaboy D, Rubin E, Pachter L, Dubchak I. Strategies and Tools for Whole-Genome Alignments. *Genome Research, in press 2002*.
15. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S. "ProbCons: probabilistic consistency-based multiple sequence alignment." *Genome Research* 15:330–340, 2005.
16. Eddy SR. Multiple alignment using hidden Markov models. *In Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pp. 114-120, 1995.
17. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. **32**(5): 1792-1797, 2004.
18. Falquet L, Pagni M, Bucher P, Hulo N, Sigrist CJ, Hofmann K, Bairoch A. The PROSITE database, its status in 2002. *Nucleic Acids Research* 30:235-238, 2002.
19. Gotoh O. A weighting system and algorithm for aligning many phylogenetically related sequences. *CABIOS* 11:543-551, 1995.
20. Gotoh O. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *Journal of Molecular Biology* 264:823-838, 1996.
21. Hardison R., Chao K.-M, Adamkiewicz M., Price D., Jackson J., Zeigler T., Stojanovic N. and W. Miller. Positive and negative regulatory elements of the rabbit epsilon-globin gene revealed by an improved multiple alignment program and functional analysis. *DNA Sequence* 4:163-176, 1993.
22. Kececioglu JD, Starrett D. Aligning Alignments Exactly. Proceedings of the 8th ACM Conference on Computational Molecular Biology (RECOMB), 28-96, 2004.
23. Kececioglu JD, Zhang W. Aligning Alignments. *In Proceedings of the 9th Annual Symposium on Combinatorial Pattern Matching* (CPM 1998).
24. Kellis M, Patterson N, Endrizzi M, Birren B, Lander E. Sequencing and comparison of yeast species to identify genes and regulatory motifs. *Nature* 423:241-254, 2003.
25. Korf I, Flicek P, Duan D, Brent MR. Integrating genomic homology into gene structure prediction. *Bioinformatics* 1:1-9, 2001.
26. Liu Y, Liu XS, Wei L, Altman RB, Batzoglou S. Conservation of eukaryotic regulatory elements and their identification using comparative genomics. *Genome Research* 14:451-458, 2004.
27. Loots GG, Ovcharenko I, Pachter L, Dubchak I, Rubin EM. rVista for comparative sequence-based discovery of transcription factor binding sites. *Genome Research* 12:832-839, 2002.
28. Ma B, Wang Z, Zhang K. Alignment between two multiple alignments. Proceedings of the 14th Symposium on Combinatorial Pattern Matching (CPM), Lecture Notes in Computer Science 2676:254-265, 2003.

29. Miller W. Comparison of genomic sequences: solved and unsolved problems. *Bioinformatics* **17:**391-397, 2000.
30. Morgenstern B, Frech K, Dress A, Werner T. DIALIGN: Finding local similarities by multiple sequence alignment. *Bioinformatics* 14:290-294, 1998.
31. Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for multiple sequence alignments. Journal of Molecular Biology 302:205-217, 2000.
32. Pollard DA, Bergman CM, Stoye J, Celniker SE, Eisen MB. Benchmarking tools for the alignment of functional noncoding DNA. BMC Bioinformatics 5:6, 2004.
33. Rivas E, Eddy SR. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* 2:8, 2001.
34. Schwartz S, Kent WJ, Smit A, Zhang Z, Baertsch R, Hardison RC, Haussler D, Miller D. Human–Mouse alignments with BLASTZ. *Genome Research* **13:**103–107, 2003.
35. Siepel A and Haussler D (2003). Combining phylogenetic and hidden Markov models in biosequence analysis. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB 2003)*, pp. 277-286.
36. Sonnhammer ELL, Eddy SR, Birney E, Bateman A, Durbin R. Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Research*, 26(1):320-322, 1998.
37. Stoye J., Evers D, Meyer F: Rose: generating sequence families. *Bioinformatics* 14:157-163, 1998.
38. Taylor W. Multiple sequence alignment by a pairwise algorithm. *CABIOS* 3:81–87, 1987.
39. Thomas JW, Touchman JW, Blakesley RW, Bouffard GG, Beckstrom-Sternberg SM, Margulies EH, Blanchette M, Siepel AC, Thomas PJ, McDowell JC. et al. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature* **424:** 788-793, 2003.
40. Thompson J.D., Higgins D.G., and T.J. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22:4673–4680, 1994.
41. Wang L, Jiang T. On the complexity of multiple sequence alignment. *Journal of Computational Biology* 1:337-348, 1994.
42. Xie Xiaohui, Liu Jun, Kulbokas EJ, Golub T, Mootha V, Lindblad-Toh K, Lander E, Kellis M. Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals. Nature 2005 Feb 27, doi:10.1038/nature03441.
43. Zhang L, Pavlovic V, Cantor CR, Kasif S. Human-mouse gene identification by comparative evidence integration and evolutionary analysis. *Genome Research* 13:1190-1202, 2003.