Aligning Sequences with Non-Affine Gap Penalty: PLAINS Algorithm, a Practical Implementation, and its Biological Applications in Comparative Genomics

Ofer Gill[1,4], Yi Zhou[3] and Bud Mishra[1,2]

[1] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA 10012.
[2] Department of Cell Biology, NYU School of Medicine, 550 First Avenue, New York, NY 10016.
[3] Department of Biology, New York University, 100 Washington Squre East, New York, NY 10003.
[4] **Corresponding Author:** gill@cs.nyu.edu; Phone: 212-998-3351

ABSTRACT **In this paper, we consider** PLAINS**, an algorithm that provides efficient alignment over DNA sequences using piecewise-linear gap penalties that closely approximate more general and meaningful gap-functions. The innovations of** PLAINS **are fourfold. First, when the number of parts to a piecewise-linear gap function is fixed,** PLAINS **uses linear space in the worst case, and obtains an alignment that is provably correct under its memory constraints, and thus has an asymptotic complexity similar to the currently best implementations of Smith-Waterman. Second, we score alignments in** PLAINS **based on important segment pairs; optimize gap parameters based on interspecies alignments, and thus, identify more significant correlations in comparison to other similar algorithms. Third, we describe a practical implementation of** PLAINS **in the Valis multi-scripting environment with powerful and intuitive visualization interfaces, which allows users to view the alignments with a natural multiple-scale color grid scheme. Fourth, and most importantly, we have evaluated the biological utility of** PLAINS **using extensive lab results; we report the result of comparing a human sequence to a fugu sequence, where** PLAINS **was capable of finding more orthologous exon correlations than similar alignment tools.**

1 INTRODUCTION

To a rough approximation, DNA sequence alignment problem differs marginally from protein sequence alignment problem. (For instance, at a superficial level, one may note that DNA alignment is over an alphabet of 4 letters whereas protein alignment is over an alphabet of 20 letters). However, two key differences are that (1) there are 3 bp DNA code per amino acid, and that (2) genes in DNA sequences that ultimately get transcripted and translated into proteins can be separated by intergenic regions of few thousands of base pairs that do not get expressed, and perhaps, are subject to strikingly different (or no) selection constraints. Thus these intergenic regions typically vary to a greater extent in one species compared to another. Therefore, we may expect the gap lengths in DNA alignments to be larger, more variable, and have specie-specific distributions. Moreover, these distributions characterizing the gap-lengths may not be memory-less (i.e., exponential distributions). There have been suggestions that power-law distributions may be more appropriate. The evolutionary processes governing the genomes of species, and the log-likelihood of certain indel gaps occurring when comparing one species against another suggest that a logarithmic

gap function is more appropriate for DNA sequences. Because of this, the traditional affine (or linear) gap functions used for aligning proteins are unsatisfactory for DNA sequences, as the ultimate results may be biologically misleading.

In order to exploit the fidelity of general non-linear gap functions for DNA sequences, without suffering performance penalites associated with them, we have chosen to use piecewise-linear gap functions modeled to approximate the gap functions in a dynamic programming approach. Here, we present an implementation of an alignment algorithm that uses reasonable amount of memory, avoids a major shortcoming associated with generalized gap penalties, and only demands a loss of constant factor (of ≤ 5.6) in time complexity compared to the best algorithm using an affine-gap model. There have been other algorithms that also proposed piece-wise linear gap model (see Miller-Myers [9]), but we present several additional theoretical innovations in terms of worst-case upper-bound memory usage, alignment optimization, and visualization of data. We have the algorithm available in a powerful bioinformatic environment, called *Valis*. Our algorithm uses an innovative learning-heuristic to determine the best score function, a near-optimal gap-penalty model, and a scheme to compute P-values for reporting alignment reliabilities.

As we hope to demonstrate here by an extensive set of experimental results, our algorithm works satisfactorily for DNA sequences, and can better reveal the underlying biological significances than other existing algorithms (e.g., needle, swat, emboss, etc.). As a concrete example, we present our alignment results for the genomic sequences of a pair of orthologous genes in Human and Fugu. While all the alternative alignment algorithms either fail by mis-aligning the exons in the Fugu sequence, or by not identifying important correlations, PLAINS is able to recover the orthologous relation between exons in the Fugu and Human sequences with good reliability. (See Fig. 2)

2 The Main Algorithm Overview

2.1 Creating an Alignment

Using a mismatch penalty ms, a p-part piecewise-linear gap penalty function $ww(\cdot)$, and reward per match fixed at 1, PLAINS generates an alignment for two sequences X and Y of lengths m and n using a method similar to Miller-Meyers [9], except that because PLAINS exclusively uses piecewise-linear gap functions (as opposed to general gap functions), it is able to take advantage of an algorithm of its very own, and uses $O(np)$ space in the worst-case[5]. Further details of how PLAINS generates an alignment, and the proofs of its $O(np)$ space bound and the correctness of the computed alignment obtained can be found in the Unabridged PLAINS paper, which can be obtained from the authors upon request.

2.2 Plains Log Approximation and Parameter Optimization

PLAINS is capable of converting any log function of log-base α and y-intercept β into a d-approximate p-part piecewise-linear function. Hence, for fixed p, a set of gap/mismatch paramters is dictated by $v = (\alpha, \beta, d, ms)$.

[5] For all practical purposes, p is fixed at some constant value ≤ 10, and hence we can say that PLAINS uses worst-case linear space.

Let $f(v)$ denote the R-score (explained later) resulting from aligning X and Y with gap/mismatch parameters taken from v. At the user's request, PLAINS can find the v to optimize $f(v)$ using either Simulated Annealing or Genetic Algorithm. Both are explained in [4]. Empirical runs over PLAINS have shown that Simulated Annealing yields better results, but Genetic Algorithm explores the space of v more thoroughly. However, all of this should come of no surprise, since (1) Monte Carlo related methods are successful in optimizing Hidden Markov Models (which are similar to sequence alignments), and (2) Genetic Algorithms typically consider subsequent solutions in a more random manner than Simulated Annealing. PLAINS is designed so that any algorithm to optimize gap/match-mismatch parameters can easily be plugged in instead of these two methods; for instance, one may search parameters with a somewhat time consuming MCMC approach, or variants such as Gibbs sampler or EM.

2.3 THE PLAINS SCORING SYSTEM

For a fixed set of gap/match-mismatch parameters v, PLAINS creates an alignment and scores it by identifying segment pairs that yields "good" scores. The sum of these segment pairs' scores (which we call R) is the value PLAINS reports, and not the score obtained from the dynamic table. This approach results in a more meaningful result, because, when we observe an alignment, it is the segment pairs of high scores that are the only true items of significance.

There are many ways to optimally select segment pairs. The method chosen by PLAINS creates the alignment first, and uses the alignment to obtain the segment pairs. Unlike nost other alignment algorithms, PLAINS avoids selecting segment pairs from the dynamic table, because PLAINS uses linear space in memory, and "grabbing" segment pairs from the dynamic table would typically require quadratic space in some way or another. Furthermore, PLAINS assumes that both mismatches and gaps are allowed in the segment pairs (though sparingly, since the segment pair has to be considered a "good" one).

Using fixed constants W, ω, and ρ, PLAINS iterates over an alignment A, collecting scores from a moving window of size W. This identifies the segments with scores of a certain percentage above the average[6] in A. Next, we trim the ends of these segments to begin and end with a match. Then, we merge any overlapping segments, recompute the scores for all segments, and get each segment's probability P'_S of having its score S occur by coincidence. Any segments with $P'_S > \rho$ are removed. Any segments that remain are the "good" ones. For those r "good" segments that remain, we conclude by computing R and ζ, where R is the sum of the scores of our r "good" segments, and ζ is the Karlin-Atschul coincidental probability that r "good" segments have a total score at least R. A further explanation for P'_S and ζ are mentioned in appendix B.

Setting $W = 50$, $\omega = 0.5$, $\rho = 0.5$ empirically yields segments that are reasonably long, and have significantly higher matches than the alignment "background". With these values for W and ω, we ran PLAINS over 900 pairs of randomly generated sequences, each with lengths ranging from 500 thru 4000, and observed the "good" segments computed.[7]

[6] Note: ω dictates this percentage. Furthermore, the choice to use a percentage above average instead of a fixed-constant cutoff score gives PLAINS the flexibility to catch important regions from any two sequences, regardless of how homologous they are to each other.

[7] Note: We temporarily ommitted the ρ-filtering step here.

From this empirical study, it was estimated that $K = 3.31 * 10^{-4}$ and $\lambda = 0.0762$, where K and λ are the Karlin-Altschul constants used to normalize segment scores.

Using our known $K, \lambda, W, \omega, \rho$ values, PLAINS can take any prespecified gap/mismatch paramters v, and report the overall alignment A obtained for sequences X and Y, along with all the "good" segment pairs, their scores and P values, and the R score and ζ value. In the event that $r = 0$, then $R = 0$ and PLAINS will report that no "good" segments were obtained.

2.4 THE PLAINS COLORGRID METHOD

For visualization of the computed alignments, the PLAINS program ported in Valis uses a coloring grid to summarize high and low matched areas for X found in the alignment. It works as follows: For some M, we color in a grid with at most M spots. We set color spot 1 based on the match percentage found in $X[1, \ldots, m/M]$ in the alignment; we set spot 2 to a color based on the match percentage found in $X[m/M + 1, \ldots, 2m/M]$ in the alignment; we set spot i to a color based on the match percentage found in $X[(i-1)m/M + 1, \ldots, im/M]$ in the alignment; and so on. The coloring grid for Y works in a similar way.[8]

Notice how here, the number of match percentages found is a fixed size. The color computations in this way has many advantages, such as how it handles the limited resolution of the computer screen compared to the sizes of X and Y.

In addition to visualizing color grids for all of X and Y, users also have the option to view portions of X or Y by specifying a substring range for either X or Y, with the Colorgrid of the unspecified sequence automatically resized to represent the corresponding area in the specified sequence's substring.

3 EMPIRICAL RESULTS

Two set of experiments were performed and used to compare PLAINS to the similar localized DNA alignment tools of EMBOSS, LAGAN, and LALIGN. The first set of experiments involve related sequences. The second set involve unrelated sequences. Tables 1 and 2 explain details and results on both sets of test runs, and appendix A features tables and figures that elaborate further. LAGAN uses piecewise-linear gap functions just like PLAINS, whereas EMBOSS and LALIGN use linear gap functions. Figure 1 shows the gap functions for PLAINS and the other tools.

Based on the results, we see that for the humanHomol runs, where there are small gaps and an expectation of medium to high homology levels, PLAINS and the other tools give close and consistent results. For the HumanPseudo runs, where there are slightly larger gaps and lower expected homology levels, PLAINS begins to show its advantages over the other tools. For the HFugu2r and HFortho runs, where there are very large gaps and very low homology levels, but still a biological relation, PLAINS is the only tool that can identify importances in such relations. Figures 2 and 3 further compare results for the HFortho2 and HumanPseudo5 runs.[9]

[8] Figures 2 and 3 are examples of this, with brighter colors signifiying high-match areas, and darker colors signifying lower-match areas. Black is used to signify areas of little-to-no match, as well as nucleotides of X or Y that were unaligned.

[9] The latter of these two figures is in appendix A.

Name	Species	Lengths	Sequence Nature	Gap Size	E(Id%)	PLAINS r	PLAINS ζ'	EMBOSS r	EMBOSS ζ'	LAGAN r	LAGAN ζ'	LALIGN r	LALIGN ζ'
humanHomol_15	Human vs.	8K vs.	cDNA	small	$86\%(nt)$	1	0.894	1	0.686	1	0.745	1	0.686
humanHomol_16	Mouse	400-3000			$90\%(nt)$	1	68.343	2	48.256	1	64.370	4	32.441
MousePseudo1	Mouse	2.4K-9.6K			$62\%(nt)$	2	3.168	2	3.087	2	3.169	1	2.965
MousePseudo2	vs.	vs.	genomic	medium	$55\%(nt)$	1	0.562	2	0.784	1	0.658	1	0.650
MousePseudo3	Pseudogene	400-500			$56\%(nt)$	0	X	0	X	0	X	0	X
HumanPseudo1					$83\%(nt)$	3	1.881	3	1.437	1	0.470	1	0.502
HumanPseudo2	Human	1.5K-11.2K			$74\%(nt)$	3	1.372	0	X	1	0.332	0	X
HumanPseudo3	vs.	vs.	genomic	medium	$85\%(nt)$	8	21.827	5	11.998	8	16.858	2	5.415
HumanPseudo4	Pseudogene	400-4000			$74\%(nt)$	1	0.538	0	X	0	X	0	X
HumanPseudo5					$75\%(nt)$	3	2.909	1	0.901	1	1.046	1	0.689
HFugu2r					$58\%(aa)$	2	2.523	0	X	0	X	0	X
HFortho1	Human	6K-12K			$55\%(aa)$	2	1.381	0	X	0	X	0	X
HFortho2	vs.	vs.	genomic	large	$52\%(aa)$	2	6.158	1	3.808	1	3.182	0	X
HFortho3	Fugu	1.8K-3.6K			$64\%(aa)$	4	3.894	0	X	0	X	0	X
HFortho4					$52\%(aa)$	4	6.071	0	X	0	X	0	X
HFortho5					$73\%(aa)$	1	3.328	0	X	0	X	0	X

Table 1. Sequence Descriptions and Results for the Related Experiments Ran. All the sequences are retrieved from ENSEMBL database [www.ensembl.org]. Note for the E(Id%) column: E(Id%) stands for Expected Identity Percentage in the Homologous Regions, and (nt) indicates match percentage of nucleotides, and (aa) indicates match percentage of amino acids after the two DNA sequences are transcribed into proteins. Shown here are the r, and ζ' values obtained from the "good" segments of each run (where $\zeta' = -log(\zeta)$). The alignment-evaluation criteria of PLAINS was used to evaluate the alignments of the other tools. Because of the triviality in arbitrarily generating high R scores, we instead look for higher ζ' values to indicate better quality alignments. When no "good" segments are found, an X is placed as the ζ' value.

Name	Species	Lengths	Sequence Nature	PLAINS r	PLAINS ζ'	EMBOSS r	EMBOSS ζ'	LAGAN r	LAGAN ζ'	LALIGN r	LALIGN ζ'
HFncd1	Human	10K-20K		0	X	0	X	0	X	0	X
HFncd2	vs.	vs.	noncoding	0	X	X	X	0	X	0	X
HFncd3	Fugu	6K-10K		0	X	X	X	0	X	0	X
FFcd1				0	X	0	X	0	X	0	X
HFcd1	Human, Fugu,	1.5K-4.8K		0	X	0	X	0	X	0	X
HHcd1	and Mouse	vs.	coding	1	1.568	0	X	0	X	0	X
HMcd1	(All six combinations)	1.5K-4.8K		2	3.517	0	X	0	X	0	X
MFcd1				0	X	0	X	0	X	0	X
MMcd1				0	X	0	X	0	X	0	X

Table 2. Sequence Descriptions and Results for the Unrelated Experiments Ran. All the sequences are retrieved from ENSEMBL database [www.ensembl.org]. Note: In the HFncd2 and HFncd3 experiments, an X is placed as the r and ζ' values for EMBOSS because EMBOSS ran out of memory aligning those experiments. Here, the unrelated sequences show that the sensitivity of PLAINS is not caused by compromised specificity. Furthermore, although not shown here, PLAINS, EMBOSS, LAGAN, and LALIGN all catch no correlations when given randomly generated DNA sequences of lengths up to 8000. Please note that the correlations that PLAINS caught in the HHcd1 and HMcd1 experiments are protein codon homologies, most of them being a short stretch of perfect matches located relatively close to each other. Although these runs were meant to check how PLAINS behaves with unrelated sequences, the correlations PLAINS caught could ironically hold some sort of importance that has been usually ignored.

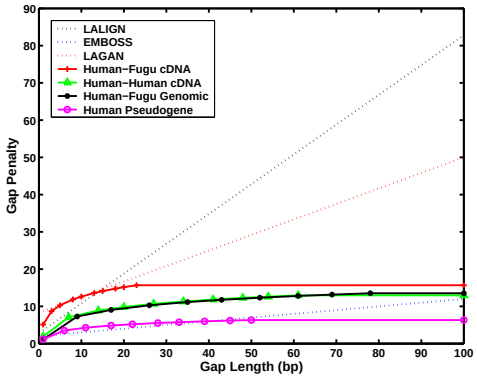

Fig. 1. Gap-parameters Graph. PLAINS optimizes the best gap/mismatch parameters based on the pair of species aligned, and the nature of the sequence. This is resemblant of LAGAN's techniques to account for the nature of species in performing its alignments. Graphed here are some of the gap-parameters PLAINS found, along with the gap-parameters of the other tools, rescaled under the assumption of 1 point per match. Note: LAGAN uses a number of unknown piecewise-linear gap parameters in aligning on a species by species basis. Shown here is simply the known default gap-parameters for LAGAN.

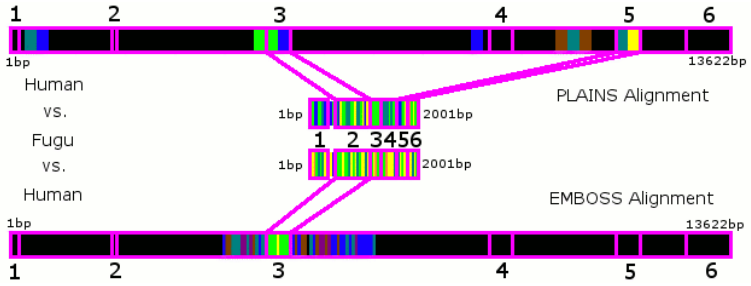

Fig. 2. Match Ratio Color Lines in the HFOrtho2 test for PLAINS and EMBOSS. Here, the Human and Fugu sequence used have six exon regions, most with some sort of correspondence to each other. Here, both PLAINS and EMBOSS correctly identify the correlation of exon region 2 in Fugu with exon region 3 in Human, but only PLAINS correctly identifies the correlation of exon region 5 in Fugu with exon region 5 in Human.

Each run of PLAINS to optimize gap/mismatch parameters on a pair of species took 30 minutes to 2 hours. The relatively long time taken by PLAINS is due to its need for computing several hundred alignments under various gap/mismatch parameters before deciding which gap/mismatch parameters are the most optimal. When ran using fixed-set gap-mismatch parameters, PLAINS ran in just under a minute, a constant factor of at most 5.6 times slower than EMBOSS. The reason for this slowdown is manifold: (1) PLAINS uses a linear space table instead of the quadratic space typical of dynamic programming, and (2) there is constant extra overhead in using Linked-List Assistance (similar to that of [9]) to help create an alignment.

Plains can easily align a pair of sequences, each with nucleotides of up to $8Kb$. It can either (1) seek the best gap-mismatch parameters for a given pair of sequences and align with those parameters, or (2) use a user-specified set of gap-mismatch parameters to align the pair of sequences. In (1), the runtime typically ranges from 30 minutes to 2 hours. In (2), the runtime typically ranges from 10 seconds to 1 minute. Plains can either

be used via commandline, or as part of the Valis tool set. More information can be found at `http://bioinformatics.nyu.edu/~gill`

4 CONCLUSIONS

PLAINS is able to catch more important correlations than its competition, especially in sequences with expected large gaps and low homologies like Human and Fugu. Furthermore, PLAINS is also capable of distinguishing related sequence pairs from unrelated ones. Consequently, we believe PLAINS is a promising tool for alignment of long regions of genomes.

References

1. Altschul, S.F., Boguski, M.S., Gish, W., and Wooton, J.C., "Issues in Searching Molecular Sequence Databases." *Nature Genetics*, **6**:119–128, 1994.
2. Michael Brudno, Chuong Do, Gregory Cooper, Michael F. Kim, Eugene Davydov, Eric D. Green, Arend Sidow, Serafim Batzoglou, "LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA," *Genome Research*, **13(4)**:721-31, 2003 Apr.
3. Gu X, Li WH., "The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment." *J Mol Evol.*, **40(4)**:464-73, 1995 Apr.
4. Hromkovic J, "Heuristics." *Algorithms for Hard Problems, Second Edition*, **6**:439-467, 2003.
5. X. Huang and W. Miller, *Advanced Applied Mathematics*, **12**:373-381, 1991.
6. Karlin S, Altschul S.F., "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes" *Proc. Natl. Acad. Sci. USA*, **87**:2264–2268, March 1990.
7. Karlin S, Altschul S.F., "Applications and statistics for multiple high-scoring segments in molecular sequences" *Proc. Natl. Acad. Sci. USA*, **90**:5873–5877, June 1993.
8. Lipman, D.J., Altschul, S.F., and Kececioglu, J.D., "A Tool for Multiple Sequence Alignment." *Proceedings of the National Academy of Sciences USA*, **86**:4412–4415, 1989.
9. Miller, W., and Myers E.W., "Sequence Comparison with Concave Weighting Functions" *Bulletin of Mathematical Biology*, **50**:97–120, 1988.
10. Miller, W., and Myers E.W., "Optimal Alignments in Linear Space" *CABIOS*, **4**:11–17, 1988.
11. Needleman, S.B., and Wunsch, C.D., "A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins." *Journal of Molecular Biology*, **48**: 443–453, 1970.
12. Ophir R, Graur D., "Patterns and rates of indel evolution in processed pseudogenes from humans and murids." *Gene.*, **205(1-2)**: 191–202, 1997 Dec 31.
13. Pearson, W.R., "Comparison of Methods for Seqrching Protein Sequence Databases." *Protein Science*, **4**:1145–1160, 1995.
14. Pearson, W.R., "Searching Protein Sequence Libraries: Comparison of the Sensitivity and Selectivity of the Smith Waterman and FASTA algorithms." *Genomics*, **11**: 635–650, 1991.
15. Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., "Downhill Simplex Method in Multidimensions." *Numerical Recipes: The Art of Scientific Computing*, **10.4**: 289–293, 1986.
16. Rice P, Longden I, Bleasby A., "EMBOSS: the European Molecular Biology Open Software Suite" *Trends Genetics*, **Jun 16(6)**:276-7, 2000.
17. Smith, T.F., and Waterman, M.S., "Identification of Common Molecular Subsequences." *Journal of Molecular Biology*, **147**: 195–197, 1981.
18. Shpaer, E., Robinson, M., Yee, D., Candlin, J., Mines, R., and Hunkapiller, T., "Sensitivity and Selectivity in Protein Similarity Searches: A Comparison of Smith-Waterman in Hardware to BLAST and FASTA." *Genomics*, **38**: 179–191, 1996.
19. States, D.J., Gish, W., and Altschul, S.F., "Basic Local Alignment Search Tool." *Journal of Molecular Biology*, **215**: 403–410, 1990.
20. Waterman, M.S., and Eggert, M., "A New Algorithm for Best Subsequence Alignments with Applications to tRNA -rRNA Comparisons." *Journal of Molecular Biology*, **197**: 723–728, 1987.
21. Zhang Z, Gerstein M, "Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes." *Nucleic Acids Res.*, **31(18)**: 5338-48, 2003 Sep 15.

Appendix

A NAMES OF THE SEQUENCES USED

Tables 3 and 4 list the specific sequences used in the experiments ran on PLAINS, EMBOSS, LAGAN, and LALIGN. Tables 5 and 6 show the R, r, and ζ' values obtained from the "good" segments of each alignment. And, figure 3 elaborates on the comparison between PLAINS and LAGAN for the HumanPseudo5 experiment.

Name	First Sequence	Second Sequence
humanHomol_15	ENST00000263253	ENSMUST00000050968
humanHomol_16	ENST00000263253	ENSMUST00000068387
MousePseudo1	ENSMUSG00000016720	pseudogene
MousePseudo2	ENSMUSG00000004038	pseudogene
MousePseudo3	ENSMUSG00000034321	pseudogene
HumanPseudo1	ENSG00000087086	pseudogene
HumanPseudo2	ENSG00000164104	pseudogene
HumanPseudo3	ENSG00000079432	pseudogene
HumanPseudo4	ENSG00000135486	pseudogene
HumanPseudo5	ENSG00000101210	pseudogene
HFugu2r	ENSG00000111845	SINFRUG00000137119 (reverse-complement)
HFortho1	ENSG00000183628	SINFRUG00000128815
HFortho2	ENSG00000099937	SINFRUG00000140660
HFortho3	ENSG00000142168	SINFRUG00000132716
HFortho4	ENSG00000138764	SINFRUG00000152968
HFortho5	ENSG00000057757	SINFRUG00000123004

Table 3. Sequence Details for the Related Experiments Ran. All the sequences are retrieved from ENSEMBL database [www.ensembl.org].

Name	First Sequence	Second Sequence
HFncd1	Human NCBI34:1:190164774:190174772	FUGU2:scaffold_5343:1:5999:1
HFncd2	Human NCBI34:22:32724006:32744004:1	FUGU2:scaffold_3421:1:9999:1
HFncd3	Human NCBI34:10:56721585:56731583:1	FUGU2:scaffold_1415:1:6999:1
FFcd1	SINFRUT00000127255	SINFRUT00000165154
HFcd1	ENSG00000150967.3	SINFRUT00000127255
HHcd1	ENSG00000150967.3	ENST00000259748
HMcd1	ENSG00000150967.3	ENSMUST00000025930
MFcd1	ENSMUST00000025930	SINFRUT00000165154
MMcd1	ENSMUST00000031354	ENSMUST00000024034

Table 4. Sequence Details for the Unrelated Experiments Ran. All the sequences are retrieved from ENSEMBL database [www.ensembl.org].

Test Name	PLAINS			EMBOSS			LAGAN			LALIGN		
	R	r	ζ'	R	r	ζ'	R	r	ζ'	R	r	ζ'
humanHomol_15	116.871	1	0.894	110.000	1	0.686	112.000	1	0.745	110.000	1	0.686
humanHomol_16	2184.956	1	68.343	1751.000	2	48.256	2064.917	1	64.370	1568.200	4	32.441
MousePseudo1	267.956	2	3.168	265.200	2	3.087	268.000	2	3.169	166.000	1	2.965
MousePseudo2	106.319	1	0.562	203.200	2	0.784	109.667	1	0.658	109.400	1	0.650
MousePseudo3	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X
HumanPseudo1	281.992	3	1.881	255.200	3	1.437	83.250	1	0.470	84.400	1	0.502
HumanPseudo2	281.851	3	1.372	0.000	0	X	89.000	1	0.332	0.000	0	X
HumanPseudo3	1782.646	8	21.827	1066.700	5	11.998	1383.583	8	16.858	439.200	2	5.415
HumanPseudo4	110.620	1	0.538	0.000	0	X	0.000	0	X	0.000	0	X
HumanPseudo5	446.425	3	2.909	138.000	1	0.901	142.667	1	1.046	131.000	1	0.689
HFugu2r	322.784	2	2.523	0.000	0	X	0.000	0	X	0.000	0	X
HFortho1	282.933	2	1.381	0.000	0	X	0.000	0	X	0.000	0	X
HFortho2	452.657	2	6.158	234.600	1	3.808	215.667	1	3.182	0.000	0	X
HFortho3	627.357	4	3.894	0.000	0	X	0.000	0	X	0.000	0	X
HFortho4	737.478	4	6.071	0.000	0	X	0.000	0	X	0.000	0	X
HFortho5	217.274	1	3.328	0.000	0	X	0.000	0	X	0.000	0	X

Table 5. Related Run Scores for PLAINS, EMBOSS, LAGAN, and LALIGN.

Test Name	PLAINS			EMBOSS			LAGAN			LALIGN		
	R	r	ζ'	R	r	ζ'	R	r	ζ'	R	r	ζ'
HFncd1	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X
HFncd2	0.000	0	X	X	X	X	0.000	0	X	0.000	0	X
HFncd3	0.000	0	X	X	X	X	0.000	0	X	0.000	0	X
FFcd1	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X
HFcd1	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X
HHcd1	139.604	1	1.568	0.000	0	X	0.000	0	X	0.000	0	X
HMcd1	341.872	2	3.517	0.000	0	X	0.000	0	X	0.000	0	X
MFcd1	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X
MMcd1	0.000	0	X	0.000	0	X	0.000	0	X	0.000	0	X

Table 6. Unrelated Run Scores for PLAINS, EMBOSS, LAGAN, and LALIGN.

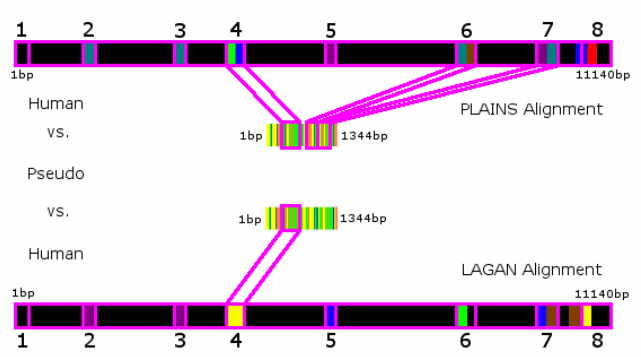

Fig. 3. Match Ratio Color Lines in the HumanPseudo5 test for PLAINS and LAGAN. Here, the Human sequence has 8 exon regions that are similar to areas of the pseudosequence used, and alignments of PLAINS and LAGAN for these cases are similar, even by eyeglance of the ColorGrids. Note that although PLAINS and LAGAN catch most of these regions in their alignments, we're only counting the exon regions that participate in "good" segments. With this in mind, PLAINS and LAGAN both identify exon region 4 as important, but PLAINS also deems exon regions 6 and 7 in the Human sequence as important, which LAGAN misses.

B THE PLAINS PROBABILITY METHOD

The PLAINS approach to estimating the significance of an alignment under an arbitrary gap model is based on the Karlin-Altschul methods from [6] and [7], and was motivated by the desire to identify the biological relevance of a generated alignment instead of just creating an arbitrary alignment with a set of "good" segments. Their methods provide a way to approximate reliability without requiring excessive biological information from our two sequences X and Y.

The Karlin-Altschul method works on gapless local alignments as follows: Suppose for each letter i that p_i is the probability of observing letter i in sequence X, and for each letter j that p'_j is the probability of observing letter j in sequence Y, and that the score for pairing letter i with j is s_{ij}. We may suppose that for a random pair of sequences, the expected alignment score $\Sigma_{i,j} p_i p'_j s_{ij}$ is negative; and nonetheless, it is possible to generate a positive score. Also, suppose each high-scoring segment is found independently of each other. Then, for some λ, $\Sigma_{i,j} p_i p'_j e^{\lambda s_{ij}} = 1$. Also, the maximum segment score, taken from a large number of independently identically distributed random variables of segments, tends to have a normal distribution. In cases of normal distribution, the probability a segment scores at least S can be approximated by e^{-S}. However, because the score increases logarithmically in terms of mn, the product of the lengths of X and Y, and the rate of this increase is unknown, and since we would also like to account for λ in our formulation, we substitue S' instead of our score S, where $S' = \lambda S - ln(Kmn)$, and hence we say that P_S, the probability a segment scores at least S is $e^{-S'} = e^{-(\lambda S - ln(Kmn))} = e^{-\lambda S + ln(Kmn)} = Kmne^{-\lambda S}$. From this, the probability P'_S that a segment of score S occurs by coincidence is also known as the probability of observing one or more segments of score S, which can be approximated as $1 - exp(-P_S) = 1 - exp(-Kmne^{-\lambda S})$.

Building under this assumption, suppose we find r highest-scoring distinct segments of scores $S_1, S_2, \ldots, S_r$, and suppose that for each k, $S'_k = \lambda S_k - ln(Kmn)$. Also assume that

$R = S_1 + S_2 + \ldots + S_r$ and $T = S'_1 + S'_2 + \ldots + S'_r$ (and hence $T = \lambda R - r * ln(Kmn)$). Then, ζ, the coincidental probability of observing r segments whose scores total to at least R, can be approximated for large T as $\frac{e^{-T}T^{r-1}}{r!(r-1)!}$.

Note that everything stated here is built under the assumptions that we are dealing with local alignments using little or no gaps, and that the high-scoring segments are obtained from the dynamic programming table. However, empirical observations have shown that these results also work when using high-scoring segments obtained directly from local alignments with significantly larger and more varied gaps, i.e., the experiments and metholodogy used by PLAINS. The K and λ values were calibrated for PLAINS based on observations of higest-scoring segments on alignments of randomly generated sequences of lengths randing from 500 to 4000.