SCORING PAIRWISE GENOMIC SEQUENCE ALIGNMENTS

F. CHIAROMONTE
Department of Statistics, Penn State,
University Park, PA 16802
chiaro@stat.psu.edu

V.B. YAP
Department of Statistics, UC Berkeley,
Berkeley, CA 94720
yapvb@stat.berkeley.edu

W. MILLER
Department of Computer Science and Engineering, Penn State,
University Park, PA 16802
webb@bio.cse.psu.edu

The parameters by which alignments are scored can strongly affect sensitivity and specificity of alignment procedures. While appropriate parameter choices are well understood for protein alignments, much less is known for genomic DNA sequences. We describe a straightforward approach to scoring nucleotide substitutions in genomic sequence alignments, especially human-mouse comparisons. Scores are obtained from relative frequencies of aligned nucleotides observed in alignments of non-coding, non-repetitive genomic regions, and can be theoretically motivated through substitution models. Additional accuracy can be attained by down-weighting alignments characterized by low compositional complexity. We also describe an evaluation protocol that is relevant when alignments are intended to identify all and only the orthologous positions. One particular scoring matrix, called *HOXD70*, has proven to be generally effective for human-mouse comparisons, and has been used by the PipMaker server since July, 2000. We discuss but leave open the problem of effectively scoring regions of strongly biased nucleotide composition, such as low G+C content.

1 Introduction

Most sequence alignment programs employ an explicit scheme for assigning a score to every possible alignment. This provides the criterion to prefer one alignment over another. Alignment scores typically involve a score for each possible aligned pair of symbols, together with a penalty for each gap in the alignment. For protein alignments, the scores for all possible aligned pairs constitute a 20-by-20 *substitution matrix*. Amino acid substitution scores are well understood in theory [2,3], and the scores most used in practice are the PAM matrices of Dayhoff [7,13] and the newer BLOSUM series [16]. The landmark studies by Dayhoff and colleagues introduced "log-odds" scores, and connected

the choice of a substitution matrix with the evolutionary distance separating two sequences.

Fewer papers have dealt with scoring schemes for alignments of DNA sequences [27,6]. A sophisticated scheme based on extensive analysis of evolutionary substitution patterns in human and rodent sequences was developed by Arian Smit, and used in the initial version of the PipMaker network server [25]. This scheme utilized non species-symmetric scores (a human `A` with a mouse `C` is not scored the same as a human `C` with a mouse `A`) to account for accelerated substitution rates in rodents [20]. Moreover, the scheme provides distinct scores for each of three ranges of G+C content (the percentage of letters that are either `G` or `C`) to account for dependence of patterns of nucleotide substitution on the latter [11]. We describe a simple log-odds technique for DNA substitution scores, reminiscent of the BLOSUM approach. Gap penalties are ignored.

A major issue in developing alignment software for genomic DNA sequences is experimental evaluation [22]. It is frequently difficult to tell which of two methods performs better in practice, in part because of the scarcity of data for which a "correct answer" is known, and in part because of disagreement on what a "correct answer" means. One may try to find protein-coding regions, regions with biologically relevant functions, or simply regions that can be reliably aligned. Perhaps the most attractive goal would be to align functional regions. However, there are very few large regions (indeed, probably none) of mammalian genomic sequence where all functional segments are known, which makes it difficult or impossible to reliably measure a program's success at attaining this ideal.

A more accessible goal is to align all detectably orthologous positions (nucleotide pairs derived from the same position in the most recent common ancestral species by substitution events). Functional regions may then be identified by other programs searching the resulting alignment for segments with special properties, such as particularly high levels of conservation [30]. The *blastz* alignment program used by PipMaker [25] takes this approach. We give a protocol for evaluating alignment software of this sort.

2 Substitution Scores

Following a common approach in protein alignment, we determine nucleotide substitution scores by identifying a set of trusted aligned symbol pairs and using log-odd-ratios [8]. To find a "training set" of nucleotide pairs, i.e., the columns of trusted alignments, we align human and mouse sequences on a pre-selected human region, using a very simple alignment program and scoring scheme. We start by deleting from the human region all interspersed re-

peats and low-complexity regions, as determined by the RepeatMasker program (Smit and Green, unpublished) with default settings, and all annotated protein-coding segments. The reduced human sequence is then aligned with the full mouse sequence using a variant of the *blast* program [1,24]. Alignments are scored by match = 1, mismatch = –1, and we retain only the matched segments that occur in the same order and orientation in both species. The program computes only gap-free alignments, commonly called *high-scoring segment pairs*.

Using the resulting training set, we apply the algorithm of Figure 1. Gap-free alignments in which nucleotide identity exceeds $max\_pct = 70\%$ (say) are discarded, so as to exclude strongly conserved portions from our analysis. (This is the step most reminiscent of the BLOSUM approach.) The hope is to accurately model moderately conserved regions, with the belief that strongly conserved ones can be found with any approach.

global int $n1(1..4), n2(1..4), m(1..4, 1..4)$ (initially all zeros)

for each gap-free local alignment **do**
  if the percent identity $\leq max\_pct$ **then**
    for each column, x-over-y, of the alignment **do**
      observe(x,y)
$npairs \leftarrow n1(A) + n1(C) + n1(G) + n1(T)$
for $x \in \{A, C, G, T\}$ **do**
  $q1(x) \leftarrow n1(x)/npairs$
  $q2(x) \leftarrow n2(x)/npairs$
  for $y \in \{A, C, G, T\}$ **do**
    $p(x, y) \leftarrow m(x, y)/npairs$
for $x \in \{A, C, G, T\}$ **do**
  for $y \in \{A, C, G, T\}$ **do**
    $s(x, y) \leftarrow \log\left(\frac{p(x,y)}{q1(x) \times q2(y)}\right)$ (scale so largest entry is 100)

procedure observe(x,y)
  infer(x, y)
  infer($compl(x), compl(y)$) (for strand symmetry)
  infer(y, x) (for species symmetry)
  infer($compl(y), compl(x)$) (for strand and species symmetry)

procedure infer(x,y)
  $m(x, y) \leftarrow m(x, y) + 1$
  $n1(x) \leftarrow n1(x) + 1$
  $n2(y) \leftarrow n2(y) + 1$

Figure 1: Algorithm to determine a matrix $s(x, y)$ of nucleotide substitution scores. The complement of nucleotide x is denoted $compl(x)$.

The score of the alignment column x-over-y is the log of an "odds ratio"

$$s(x,y) = \log\left(\frac{p(x,y)}{q_1(x)q_2(y)}\right) \tag{1}$$

where $p(x,y)$ is the frequency of x-over-y in the training set, expressed as a fraction of the observed aligned pairs, and $q_1(x)$ and $q_2(y)$ denote the background frequencies of nucleotides x and y as the upper and lower components (resp.) of those same pairs. Frequencies actually include also aligned pairs "inferred" from the observed ones. For each x-over-y, we infer $compl(x)$-over-$compl(y)$, where *compl* denotes nucleotide complement (so $compl(A) = T$). This makes the scores strand symmetric, i.e., invariant under reverse complementation of the two sequences. Moreover, for each x-over-y we infer y-over-x. This makes the scores species symmetric ($s(x,y) = s(y,x)$) so that the same matrix can be used for human-mouse and mouse-human alignment, but the algorithm in Figure 1 can be used to compute two asymmetric matrices deleting the statements enforcing symmetry from the *observe* procedure. In applications, we find that species symmetric matrices work about as well as asymmetric ones (see Section 4). To permit use of integer arithmetic, we normalize the scores $s(x,y)$ so that the largest is 100, then round to the nearest integer.

Here we give substitution matrices calculated on three different human-mouse training sets. The regions were chosen to approximately span the range of G+C content seen in the human genome. In all three cases, *max_pct* was set to 70%.

	CFTR [9] matrix 37.4% G+C				HOXD matrix 47.5% G+C				hum16pter [10] matrix 53.7% G+C			
	A	C	G	T	A	C	G	T	A	C	G	T
A	67	–96	–20	–117	91	–114	–31	–123	100	–123	–28	–109
C	–96	100	–79	–20	–114	100	–125	–31	–123	91	–140	–28
G	–20	–79	100	–96	–31	–125	100	–114	–28	–140	91	–123
T	–117	–20	–96	67	–123	–31	–114	91	–109	–28	–123	100

Any of these matrices can then be used in the traditional way: to evaluate the relative likelihood that a gap-free alignment correctly matches related sequences, as opposed to unrelated ones, we read its column scores off the matrix, and add them together.

One possible refinement of this approach would be to compute a custom matrix each time sequences are aligned; that is, derive the training set from the region undergoing alignment itself. This could be easily and efficiently implemented within many existing alignment programs. For instance, the *blastz* program used by PipMaker operates in three phases, similar to those of the gapped *blast* program [4]: (1) find short exact matches, (2) determine ungapped extensions of the exact matches, (3) for sufficiently high-scoring ungapped

matches, compute alignments allowing for gaps. Step (2) is relatively inexpensive, typically taking about 10% of the execution time. An initial set of ungapped alignments can be computed with generic substitution scores and used as described above to determine a locus-specific scoring matrix. Then phase (3), perhaps preceded by an iteration of phase (2), can utilize the customized scores. One might even consider several iterations of phase (2) and re-computation of substitution scores.

Another possible refinement of our approach would be to segment long genomic regions undergoing alignment into relatively homogeneous subregions (e.g., with respect to G+C content or, more directly, with respect to patterns of substitution frequencies) and use different substitution matrices in each subregion. Lastly, our approach could be generalized to the estimation of 16-by-16 matrices accounting for dependence of nucleotide substitution on adjacent nucleotides (e.g. `CG` tending to become `TG` or `CA`, and other similar effects [17]). Precedent for this can be found also in protein sequence alignment, with 400-by-400 substitution matrices [14].

2.1 Modeling substitution

An alternative method of deriving substitution scores from a training region is to view its gap-free alignments as independent realizations of a *reversible time-continuous Markov chain*. This models the substitution process linking the segments of each gap-free alignment through a common ancestral segment. The process is characterized by a 4 by 4 rate matrix calibrated to produce on average 1% substitutions per unit time, and the segments of each gap-free alignment are separated by an alignment-specific divergence time, which roughly corresponds to the percent identity. Thus, a unique process is viewed as generating alignments with different degrees of identity through different divergence times.

Numerical maximization of the likelihood function of this model provides estimates of the rate matrix, say Q, and of the divergence times, say t_ℓ, $\ell = 1, 2, \ldots$. Q can be used to estimate frequencies and background frequencies for a generic divergence time t as:

$$\begin{aligned} p_t(x,y) &= \pi(x)\exp\{Q(x,y)t\} \\ q_{1,t}(x) = \sum_y p_t(x,y) &= \pi(x) \qquad q_{2,t}(y) = \textstyle\sum_x p_t(x,y) = \pi(y) \end{aligned} \tag{2}$$

where $\exp\{Q(x,y)t\}$ estimates the chance of y substituting x over t time units, and $\pi(x)$, $x =$ `A`, `C`, `G`, `T` the chance of x in the stationary distribution of

the process. Using these quantities in equation (1), we can compute a t-dependent scoring matrix $s_t(x, y)$. Although we could produce alignment-specific substitution matrices setting $t = t_\ell$, we produce a single matrix from the training region as follows. The frequencies $p(x, y)$ from the algorithm in Figure 1, considered as a whole, define a rate matrix $\bar{Q}$ and an "overall" divergence time $\bar{t}$. Setting $t = \bar{t}$ in equation (2) we obtain frequencies $p_{\bar{t}}(x, y)$, $q_{1,\bar{t}}(x)$, $q_{2,\bar{t}}(y)$, and scores $s_{\bar{t}}(x, y)$. Thus, when comparing these scores with the $s(x, y)$ obtained directly from $p(x, y)$, $q_1(x)$, $q_2(y)$, we are actually comparing two rate matrices, Q and $\bar{Q}$, using the same divergence time $\bar{t}$.

If we restrict attention to gap-free alignments with percent identity $\leq 70\%$, numerical likelihood maximization of the reversible Markov chain model on the CFTR, HOXD and hum16pter training regions gives scoring matrices practically indistinguishable from the ones generated by the algorithm in Figure 1, and lends a strong theoretical motivation to this simple procedure.

2.2 Score adjustment for low-complexity regions

Whatever the selected training regions and estimation procedures, the log-odds score contribution $s(x, y)$ of a pair x-over-y observed in the regions undergoing alignment is high if the pair occurs more often than by chance *in the training data*. But the compositional complexity of the regions under alignment may differ substantially from that of the training data. In particular, low compositional complexity in the regions under alignment may increase chance occurrence of pairs that are relatively rare in the training data, and hence misleadingly inflate the scores of some gap-free alignments.

A simple approach to adjust for such an effect is to multiply the score of each gap-free alignment ℓ by the relative *entropy* characterizing its top segment:

$$H(\ell) = \frac{-\sum_x q_{1,\ell}(x) \log q_{1,\ell}(x)}{\log 4}$$

where $q_{1,\ell}(x)$, $x =$ `A`, `C`, `G`, `T`, are the background frequencies in the top segment.

As $H(\ell)$ ranges in $[0, 1]$, this adjustment achieves the desired effect of down-weighting misleadingly high scoring alignments (the fact that it works counter-intuitively for low scoring ones – e.g., an alignment with negative score will have a lower adjusted score the higher its relative entropy – is inconsequential).

Since all the substitution matrices we are considering have positive entries only along the main diagonal, a high scoring alignment ℓ will have an abundance of matches: the alignment frequencies $p_\ell(x, y)$ will be largely concentrated on x-over-x pairs, and very small on mismatches; $p_\ell(x, y)$, $x \neq y$.

Consequently

$$-\sum_x q_{1,\ell}(x)\log q_{1,\ell}(x) \approx \sum_{x,y} p_\ell(x,y)\log\left(\frac{p_\ell(x,y)}{q_{1,\ell}(x)q_{2,\ell}(y)}\right)$$

so that adjusting the score of a high scoring alignment by $H(\ell)$ is approximately the same as adjusting it by its relative *expected quantity of information*. The expected quantity of information benchmarks pair occurrences in ℓ against the background frequencies of ℓ itself, so it can be interpreted as the "score of an alignment against itself": alignments that score high against the training set will be down-weighted if they score poorly against themselves. Although it is possible, in principle, to use directly the expected quantity of information instead of the entropy, the latter has the advantage of involving computations on only one of the sequences undergoing alignment.

3 Evaluation Procedure

We now describe a simple protocol for evaluating alignment software and substitution scores with respect to the goal of aligning orthologous positions.

Orthologous human and mouse genomic regions believed to be free of large-scale rearrangements such as gene duplications (small inversions shouldn't matter) differ due to nucleotide substitutions, small-scale insertions/deletions, and insertion of interspersed repeats. When the regions are aligned, the true matches appear along a diagonal path in the dot-plot, with spurious matches off the diagonal. To a first approximation, paired nucleotides on that path can be considered correct, and paired nucleotides off the path incorrect, treating overlapping alignment with some care. (The path can be determined by any of several methods [31,32].)

We again used the primitive *blast* program [24] for gap-free alignments. With each of a variety of scoring schemes, we determined the gap-free alignments that scored above various thresholds (denoted K below), then divided the aligned nucleotide pairs into correct (for the maximal chain of properly ordered matches) and incorrect (all other matches).

For instance, aligning the human CD4 region and its known mouse ortholog using the HOXD matrix (which might more properly be called HOXD70, to emphasize dependence on $max\_pct = 70\%$), we obtain the two dot-plots in Figure 2. The left panel shows alignments scoring at least 2000, and the right one those scoring above 3000. Note that increasing the threshold substantially reduces the number of spurious matches (off-diagonal), but at the cost of slightly reducing the number of putatively correct matches (on the diagonal path), a typical sensitivity-specificity tradeoff.

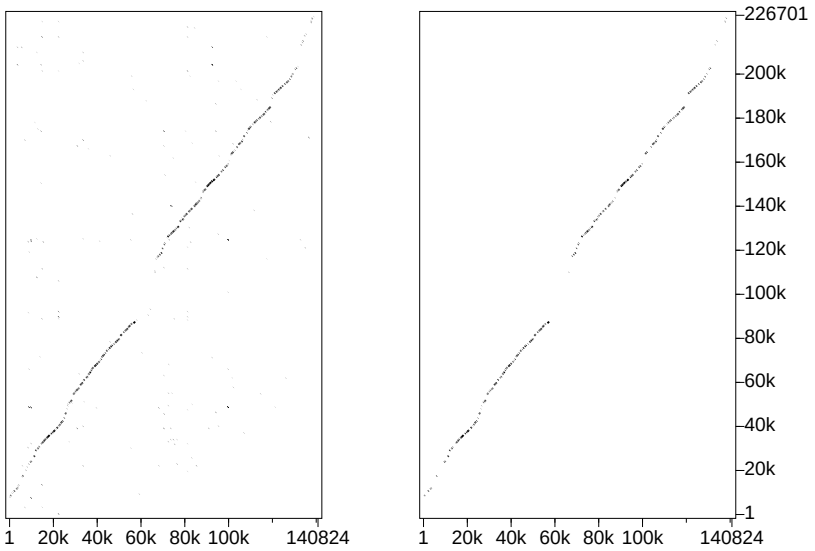

Figure 2: Dot-plots for alignments of the human and mouse CD4 loci using the HOXD matrix, for $K = 2000$ (left) and $K = 3000$ (right).

Using CD4 as our test region, the following table reports exact counts of correct and incorrect nucleotide pairs for four different scoring schemes, at six different threshold K levels - the last column refers to the HOXD matrix with subsequent entropy adjustment of alignment scores.

unit (±1)			hum16pter			HOXD			HOXD+entropy		
K	right	wrong	K	right	wrong	K	right	wrong	K	right	wrong
20	47751	7942	2000	52919	12230	2000	53021	10007	1800	54054	5863
22	45862	1690	2200	50544	4526	2200	50468	3084	2000	51646	1799
24	43378	495	2400	48370	2017	2400	48246	941	2200	49480	540
26	41614	378	2600	46403	870	2600	46416	697	2400	46997	277
28	40227	252	2800	44326	204	2800	44794	272	2600	45625	65
30	38970	34	3000	42675	65	3000	43126	101	2800	43890	65

The six K levels considered for each scheme are different because of the different maximal value of a one-column score. This is 1 for the unit matrix, 100 for the hum16pter and HOXD matrices, and about 90 after correcting HOXD scores for entropy. Thus, the rows of the table are comparable, with threshold levels corresponding to maximal scoring contiguous matches of length 20, 22, 24, etc. The same information is summarized in Figure 3, plotting correct versus incorrect counts for each scoring scheme, at the various threshold levels.

The unit matrix lies well below all others, at all threshold levels. Among

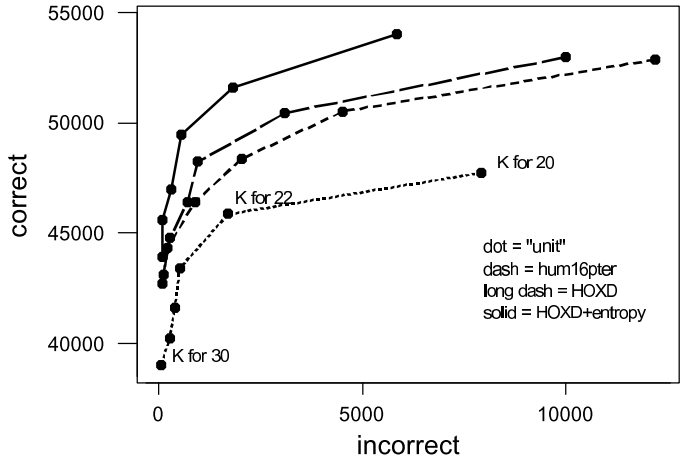

Figure 3: Correct vs. incorrect matches for human-mouse alignments of the CD4 region. Each curve corresponds to a scoring scheme, and comprises six threshold levels.

the non-unit schemes, the HOXD matrix with entropy adjustment is uniformly better, and more markedly so at low threshold levels. The HOXD and hum16pter matrices have comparable performances for high thresholds, but at low thresholds HOXD reduces the number of "false positives" with respect to the hum16pter. Thus, HOXD performs better despite originating from a region with G+C content less like that of CD4.

In addition to inspecting graphs like those described above, we summarized the comparison of two scoring schemes at a genomic locus with a single number, as follows. Consider a first scheme, say the HOXD matrix. We focus on "corner thresholds", i.e. thresholds that, if decreased by 1, produce a strictly larger number of incorrect matches. These are the relevant values because any non-corner K would be automatically discarded in favor of $K-1$, producing the same number of incorrectly aligned nucleotides and a larger or equal number of correctly aligned ones. For instance, at the CD4 locus, the HOXD matrix with $K = 2433$ gives 48796 correct versus 991 incorrect matches, while passing to $K = 2432$ gives 48796 versus 1047. Thus $K = 2433$ is a corner threshold for the HOXD matrix when aligning our CD4 sequences.

A second scoring scheme can then be compared to HOXD on its corner thresholds; exploring thresholds at which a second scoring matrix applied to CD4 produces about 1000 incorrect matches, we found 47001 correct versus

1106 incorrect at threshold 2350, and 46876 versus 981 at threshold 2351. Thus, at this corner, the HOXD matrix identifies about 1800-1900 more correct nucleotides for the same cost. We have software that performs this inspection at each corner threshold, and reports won-lost-tie counts. The won/lost ratio provides a single quantity to summarize the relative performance of two scoring schemes at a given genomic locus.

4 Experimental Results

We compared the HOXD matrix to eight other matrices, and to HOXD with the correction for entropy. The table below provides won/loss ratios on HOXD corner thresholds for nine genomic regions, each named for a gene that it contains – we always deleted interspersed and simple repeats from the human sequence (RepeatMasker with default settings). HOXD was superior on each comparison with a ratio larger than 1.

The table's second column gives the region's G+C content. Columns 3 and 4 refer to match/mismatch matrices; the unit matrix, and a match = 19, mismatch = –16 matrix suggested on theoretical grounds by Stephen Altschul as being the most appropriate match/mismatch choice for human-rodent comparisons. Columns 5 and 6 refer to the hum16pter and CFTR matrices. Then come three matrices proposed by Arian Smit for human-mouse comparisons in genomic regions of approximately 37%, 43% and 50% G+C content, respectively. The next-to-last column refers to an asymmetric version of the HOXD matrix, computed removing species and strand symmetrization from the algorithm in Fig. 1. Last comes the HOXD matrix with entropy adjustment.

Region	%G+C	±1	$\frac{+19}{-16}$	16pt	CFTR	S37	S43	S50	asym	entro
MYO15 [21]	55.1	∞	∞	1.75	2.06	0.11	0.0	0.0	0.122	0.26
CD4 [5]	51.1	∞	83.0	15.8	∞	∞	2.42	11.0	∞	0.0
MECP2 [28]	48.6	∞	∞	13.3	∞	∞	4.2	∞	0.23	0.30
CECR [12]	47.3	30.0	9.3	19.7	14.5	11.4	6.6	5.9	3.3	0.051
SCL [15]	46.4	2.0	1.2	5.2	∞	7.3	∞	∞	0.79	0.85
BTK [23]	43.2	∞	0.87	13.0	1.54	5.5	13.0	13.0	6.0	0.4
Mnd2 [18]	41.4	11.0	0.50	12.0	∞	∞	12.0	12.0	12.0	0.083
FHIT [26]	38.4	∞	58.0	2.58	∞	117.0	18.7	117.0	0.24	0.035
SNCA [29]	36.3	∞	∞	1.3	∞	∞	0.28	0.53	1.0	0.0

These results allow us to draw some conclusions, and identify some open questions. Match/mismatch scores, which ignore the higher probability of transitions (conversion between `A` and `G`, or between `C` and `T`)) with respect to transversions (any other nucleotide substitution), can be substantially improved upon: The HOXD matrix does distinctly better than the unit matrix on all test regions, and better than the $+19/-16$ matrix on most. Another

clear point is that our simple approach to down-weighting low-complexity regions improves performance: HOXD with entropy correction does distinctly better than HOXD itself on all regions.

Certain ambiguities remain. The asymmetric version of HOXD does better than HOXD on some regions, but worse on others, and the performance does not appear to be related to G+C content. HOXD does better than hum16pter on all regions, including those with high G+C, and better than CFTR on all regions, including those with low G+C. Similarly, HOXD does better than S37 on low G+C regions, better than S43 on medium G+C regions, and better than S50 on all high G+C regions except the most extreme, MYO15. On this region, though, *all* S matrices do better than HOXD. The lowest G+C region SNCA provides another seemingly paradoxical situation, since S43 and S50 do better than HOXD while S37 does worse.

Figure 3 shows how, on the CD4 test region, HOXD reduces the number of "false positives" with respect to hum16pter. While computing the won/loss ratio for the HOX–CFTR comparison on the FHIT test region, we observed that "false positives" identified by CFTR tended to have nucleotide composition different from that of "true positives" and of FHIT as a whole. For instance, at one threshold we obtained `A`:31.7%, `C`:18.2%, `G`: 19.1%, `T`: 31.0% for correct, and `A`: 39.7%, `C`:10.8%, `G`:34.0%, `T`:15.4% for incorrect alignments.

Excluding its own asymmetric and entropy corrected versions, the HOXD matrix wins 56 out of 63 comparisons. As reported by Lander *et al.* [19] (p.883), the HOX clusters are characterized by the lowest density of interspersed repeats in the human genome, making correct local alignments relatively easy to produce, even in segments with nucleotide identity below 70%. Moreover, the alignment-specific divergence times estimated with our reversible Markov chain model do not present a sizeable correlation with alignment-specific G+C content within HOXD (the correlation coefficient is 0.013, compared to 0.242 in hum16pter and -0.593 in CFTR). These factors and others may explain why local alignments from the HOXD region provide particularly effective training data for computing a single log-odds score matrix that performs well in a variety of contexts.

However, several aspects of our analysis strongly suggest that further improvements in scoring genomic DNA sequence alignments will likely be generated by exploiting G+C content and other local compositional properties.

Acknowledgments

Stephen Altschul suggested the $+19/-16$ match/mismatch matrix and David Haussler made several helpful recommendations. Our work was supported by

grant HG02238 from the National Human Genome Research Institute.

References

1. S. Altschul *et al.*, *J. Mol. Biol.* **215**, 403 (1990)
2. S. Altschul, *J. Mol. Biol.* **219**, 555 (1991)
3. S. Altschul, *J. Mol. Evol.* **36**, 290 (1993)
4. S. Altschul *et al.*, *Nucleic Acids Res.* **25**, 3389 (1997)
5. M. Ansari-Lari *et al.*, *Genome Research* **8**, 29 (1998)
6. W. Bains, *DNA Sequence-J.DNA Sequencing and Mapping* **3**, 267 (1993)
7. M. Dayhoff *et al.*, in *Atlas of Protein Sequence and Structure*, M. Dayhoff ed., p. 345, 1978.
8. R. Durbin *et al.*, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.
9. R. Ellsworth *et al.*, *Proc. Natl. Acad. Sci. USA.* **97**, 1172 (2000)
10. J. Flint *et al.*, *Human Molecular Genetics* **10**, 371 (2001)
11. M. Francino, H. Ochman, *Nature* **400**, 30 (1999)
12. T. Footz *et al.*, *Genome Research* **11**, 1053 (2001)
13. D. George *et al.*, *Methods in Enzymology* **183**, 333 (1990)
14. G. Gonnet *et al.*, *Biochem. Biophys. Res. Comm.* **199**, 489 (1994)
15. B. Gottgens *et al.*, *Genome Research* **11**, 87 (2001)
16. S. Henikoff, J. Henikoff, *Proc. Natl. Acad. Sci. USA* **89**, 10915 (1992)
17. S. Hess *et al.*, *J. Mol. Biol.* **236**, 1022 (1994)
18. W. Jang *et al.*, *Genome Research* **9**, 53 (1999)
19. E. Lander *et al.*, *Nature* **409**, 806 (2001)
20. W. Li *et al.*, *Mol Phylogenet. Evol.* **5**, 182 (1996)
21. Y. Liang *et al.*, *Genomics* **61**, 243 (1999)
22. W. Miller, *Bioinformatics* **17**, 391 (2001)
23. J. Oeltjen *et al.*, *Genome Research* **7**, 315 (1997)
24. S. Schwartz *et al.*, *Nucleic Acid Research* **19**, 4663 (1991)
25. S. Schwartz *et al.*, *Genome Research* **10**, 577 (2000)
26. T. Shiraishi *et al.*, *Proc. Natl. Acad. Sci. USA.* **98**, 5722 (2001)
27. D. States *et al.*, *METHODS: A companion to Methods in Enzymology* **3**, 66 (1991)
28. K. Reichwald *et al.*, *Mammalian Genome* **11**, 182 (2000)
29. J. Touchman *et al.*, *Genome Research* **11**, 78 (2001)
30. N. Stojanovic *et al.*, *Nucleic Acids Res.* **27**, 3899 (1999)
31. W. Wilbur and D. Lipman, *Proc. Natl. Acad. Sci. USA* **80**, 726 (1983)
32. Z. Zhang *et al.*, *J. Comput. Biol.* **1**, 217 (1994)