Statistical Identification of Uniformly Mutated Segments within Repeats

S. Cenk Sahinalp* Evan Eichler† Paul Goldberg‡
Petra Berenbrink§ Tom Friedetzky¶ A. Funda Ergun‖

Abstract

Given a long string of characters from a constant size (w.l.o.g. binary) alphabet we present an algorithm to determine whether its characters have been generated by a single i.i.d. random source. More specifically, consider all possible k-coin models for generating a binary string S, where each bit of S is generated via an independent toss of one of the k coins in the model. The choice of which coin to toss is decided by a random walk on the set of coins where the probability of a coin change is much lower than the probability of using the same coin repeatedly. We present a statistical test procedure which, for any given S, determines whether the *a posteriori* probability for $k = 1$ is higher than for any other $k > 1$. Our algorithm runs in time $O(\ell^4 \log \ell)$, where ℓ is the length of S, through a dynamic programming approach which exploits the convexity of the *a posteriori* probability for k.

The problem we consider arises from two critical applications in analyzing long alignments between pairs of genomic sequences. A high alignment score between two DNA sequences usually indicates an evolutionary relationship, i.e. that the sequences have been generated as a result of one or more duplication events followed by random point mutations. Such sequences may include functional regions (e.g. exons) as well as non-functional ones (e.g. introns). Functional regions with critical importance exhibit much lower mutation rates than non-functional DNA (or DNA with non-critical functionality) due to selective pressures for conserving such regions. As a result, given an alignment between two highly similar genome sequences, it may be possible to distinguish functional regions from non-functional ones using variations in the mutation rate. Our test provides means for determining variations in the mutation rate and thus checking the existence of DNA regions of varying degrees of functionality.

A second application for our test is in determining whether two highly similar, thus evolutionarily related, genome segments are the result of a single duplication event or of a complex series of duplications. This is particularly an issue in evolutionary studies of genome regions rich with repeat segments (especially non-functional tandemly repeated DNA). Our approach can be used to distinguish simple duplications from complex repeats again by exploiting variations in mutation rates.

*Dept of EECS, Dept of Genetics and Center for Computational Genomics, Case Western Reserve University, USA; cenk@cwru.edu. This work is supported in part by an NSF Career Award and grants from Ohio Board of Regents and Charles B. Wang Foundation.

†Dept of Genetics and Center for Computational Genomics, Case Western Reserve University, USA eee@po.cwru.edu

‡Dept of Computer Science, University of Warwick, UK; pwg@dcs.warwick.ac.uk. Partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

§Simon Fraser University, School of Computing, Canada; petra@cs.sfu.ca

¶Pacific Institute of Mathematics, Simon Fraser University, Canada; tf@pims.math.ca

‖NEC Research Institute and Dept of EECS, CWRU, USA ergun@research.nj.nec.com

1 Introduction

As more of the human genome is sequenced and assembled, it is becoming apparent that it consists of numerous sequences "repeated" with various degrees of similarity [20, 3, 11]. Such repeats are more likely to be generated as a result of segmental duplications during evolution rather than by chance.[1]

Approximately 60% of the human genome appears to be repeated.[2] Repeat segments are commonly classified into three main categories. Over 45% of the human genome comprises common repeats; one example is the ~ 300bp *alu* element, occurring more than 1M times within a divergence rate of $5\% - 15\%$. Another $\sim 5\%$ consists of the centromeric repeats, particularly the alpha satellite and microsatellite DNA. A final $\sim 7\%$ is made up of much longer repeat segments (which include partial or complete genes) exhibiting small divergence rates ($\leq 10\%$). These figures support the theory that duplications followed by point mutations provide the main process underlying genome evolution [10].

Several biochemical mechanisms underlying segmental duplications have been identified in the last 30 years (e.g. *unequal cross over* [17], *replication slippage* and *retrotransposition*); potentially many more are waiting to be discovered. The task of identifying all duplication mechanisms in the genome for a better understanding of the genome evolution process poses a number of algorithmic and computational challenges. First and foremost, one needs to identify *a posteriori* all pairs of repeat sequences which were generated as a result of a single duplication event during evolution. Note that a *repeat* can be a result of multiple complex *duplication* events: for example a long tandemly repeated sequence S, S, S, S may be a result of tandemly duplicating (copying) the first S three times or duplicating the first S once to obtain S, S and duplicating this whole segment again to get S, S, S, S; there are many other possibilities. This is the subject of our concern.

Contributions. In this paper we present a statistical test for identifying whether a pair of genome sequences with a high similarity score are indeed a result of a single duplication event. For this purpose we employ the "neutral hypothesis"; i.e. that point mutations occur independently at random with a fixed probability ($1.5 - 3 \times 10^{-9}$ per base pair per year for non-functional segments of humanoids and old world monkeys).

As mentioned above, a high similarity score between two sequences is an indication of an evolutionary relationship. One possible relationship between two such sequences is that one may have been copied from the other in a single duplication event. Because after a duplication both copies would be subject to independent point mutations, a number of edit errors would be observed in their alignment. However, these mutations would have been applied to each character in an i.i.d. fashion; as a result, the normalized similarity score between the two sequences is expected to be uniform (allowing for the statistically expected amount of variation) throughout their alignment.

A common strategy for identifying duplications between two genome regions is to iteratively locate

[1] Consider, for example, a 3×10^9bp sequence (which is roughly of size similar to the human genome) generated by an i.i.d. random source on the four letter DNA alphabet. Then the probability that two specified 100bp long segments have a Hamming distance of 5% or less is smaller than 2^{-150}. Thus, the probability that the whole genome (generated as above, without any duplication events) includes a pair of 100bp long segments with Hamming distance of 5% or less is smaller than 2^{-75} (practically nil).

[2] Studies show that 60% of the nucleoties are contained in 1Kbp segments repeated with less than 30% divergence.

pairs of sequences with the highest similarity score (e.g. via Smith-Waterman method). Shortcomings of this strategy in terms of "signal strength" are discussed in [2], where an alternative "normalized" similarity measure based on [14] is described along with an efficient algorithm for computing it. This approach is designed for identifying pairs of sequences with higher functional relationship rather than providing a tool for studying the evolution of repeat segments. As mentioned at the beginning of this section, a pair of sequences with a high alignment score may be a result of a number of complex duplication events occurring at different points of evolutionary time. They may also involve segments with varying degrees of functionality which are subject to different rates of mutation; this is due to evolutionary pressures for conserving highly functional segments. As a result, a high overall alignment score (absolute or normalized) cannot be used (due to its consolidation of the individual alignment scores of smaller segments in the sequences) to measure the evolutionary time passed since the separation of two such sequences. For instance, in satellite DNA, which contains a large number of tandem repeats of the same subsequence, there are many possibilities as to the actual progression of the duplications, including their order, as well as the source and destination subsequences [9]. Assuring that the sequences have been subject to independent point mutations only, rather than a complex series of duplication events, is critical to the accuracy of phylogenetic analysis, especially based on distance comparisons (e.g. [18]) involving these sequences.

To address the above issue we propose a new method for pairwise sequence comparison in the form of a *statistical test* to determine whether a given pair of sequences with high similarity score have been generated as the result of a single duplication event. More specifically, we consider k-state Hidden Markov Models (HMMs) for generating the alignment sequence S (on which a 0 may represent a correct alignment and a 1 may represent a misalignment) between two highly similar sequences. In the models that we consider, the bit values of S are generated by independent tosses of biased coins (with output 0/1) which are fixed for each state of the HMM in consideration. (Thus each state represents a random process which imposes a fixed mutation rate on the segment it is applied upon.) The sequence of states which are responsible for the generation of S is decided by a random walk where the probability of a state change is much lower than that of remaining at a given state. We present an algorithm which determines *a posteriori* for any given S, whether among all possible k-state Hidden Markov Models, those for which $k = 1$ are more likely than any other for which $k > 1$ (we compare the aggregate likelihood of all 1-state HMMs with that of k-state HMMs for $k > 1$). Our algorithm runs in time $O(\ell^4 \log \ell)$, where ℓ is the length of S, through a dynamic programming approach which exploits the convexity of the probability function for k.

2 Preliminaries

For the purposes of this paper, the *genome* is a long string of characters from the DNA alphabet $\{a, c, g, t\}$. A *genome segment* is a substring of the genome. We assume that we are given the correctly assembled genome (partially or as a whole) as part of the input.

Throughout the paper S and Q denote genome segments, $S[i]$ denotes the i^{th} character of segment S, and $S[i : j]$ the substring between the i^{th} and j^{th} characters (inclusive) of S. $|S|$ denotes the length of the segment S.

An alignment between two genome segments Q and S is the 2-tuple (Q', S'), where $Q', S' \in$

$\{a, c, g, t, -\}^\ell$ for some $\ell = |Q'| = |S'|$, such that Q and S are obtained if all "$-$" are removed from Q' and S' respectively.

Given two characters x and y, $x \oplus y$ denotes the character-wise exclusive-OR (XOR) function; it evaluates to 1 if $x \neq y$ and to 0 otherwise. Given an alignment (Q', S'), where $|Q'| = |S'| = \ell$, we denote by $al(Q', S') = Q' \oplus S'$, the *alignment sequence* of (Q', S') whose i^{th} entry is $Q'[i] \oplus S'[i]$. We denote by $h(Q', S')$ the normalized Hamming distance between Q' and S', i.e. the number of 1's in $al(Q', S')$ divided by ℓ.

3 A statistical test for detecting simple duplications

The sequence comparison problem we consider can be formally described as follows. We are given two genome segments Q, S and their alignment (Q', S') for which $h(Q', S') \leq \delta$ for some predetermined threshold value $0 \leq \delta \leq 1$. Our goal is to determine whether the alignment sequence $al(Q', S')$ is more likely to have been generated by a single i.i.d. random source or a combination of k i.i.d. random sources, for some $k > 1$.

The underlying motivation for the above problem is the need to test whether the sequences Q and S have been generated by a single duplication event, followed by independent point mutations only. Alternatively they could either be a result of more complex procession of multiple duplication events, where the segments involved were subjected to mutations for different periods of time, or a result of a single duplication event after which different subregions have been subjected to different mutation rates. When the latter possibility is indeed the case, one expects to observe varying mutation rates throughout the sequences, resulting in measurable variation in the normalized distance between aligned segments of Q' and S'. Thus a statistical test for determining whether the edit errors between Q' and R' are more likely to have been generated by a single i.i.d. random source than by multiple sources can be used as a tool for identifying pairs of sequences that have been a result of a single duplication event. For such sequences an overall similarity score can be used to determine the evolutionary time passed since their separation.

3.1 Comparing single and multiple coin models

Given the alignment sequence $O = al(Q', S') = (O_1, \ldots, O_\ell)$ of length ℓ, we would like to compute the *a posteriori* probability that O has been generated by independent tosses of a single coin or by a procession of multiple, coins selected by performing a random walk in the set of coins. More formally, we define an n-coin model as an n-state Hidden Markov Model similar to many other applications of HMMs [4]. Note that employing HMMs in the context of this paper is quite natural. Without any *a priori* information about which positions in the alignment sequence a coin switch is more likely, it is plausible to assume independent and identical distributions for the coin switch probabilities; this in turn defines an HMM.

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ denote a set of n coins, each with 0/1 outcome. Let $p_i(b)$ denote the probability of outcome b, $b \in \{0, 1\}$ on a flip of coin C_i. Thus, $p_i(0) + p_i(1) = 1$. If $n = 1$, we will denote C_1 as C and $p_1(1)$ as p. We denote by q_j, the coin used in generating $O[j]$, $1 \leq j \leq \ell$. Note that for any $t = 1, \ldots, \ell$, $\Pr(O[t] = 1 \mid q_t = C_j) = p_j(1)$, since the outcome does not depend on the location itself but only on the "active" coin. Furthermore, let $a_{i,j}(t) = \Pr(q_{t+1} = C_j \mid q_t = C_i)$

denote the transition probability for from coin C_i to coin C_j between locations t and $t+1$. Again, $a_{i,j}$ does not depend on the location t, hence in the following we will refer to $a_{i,j}$ only. Let $\pi_j = \Pr(q_1 = C_j)$ for $1 \leq j \leq n$ (the probability that the first location is generated by coin C_j).

With $A = \{a_{i,j} \ : \ 1 \leq i, j \leq n\}$, $P = \{p_j(1) \ : \ 1 \leq j \leq n\}$, and $\pi = \{\pi_j \ : \ 1 \leq j \leq n\}$, an n-coin model λ is now defined as $\lambda = (A, P, \pi)$. For $i \geq 1$, let Λ_i denote the set of all i-coin models. Denote by $\Lambda = \bigcup_{i \geq 1} \Lambda_i$ the set of all coin models.

Let $\Omega = \{0,1\}^\ell \times \Lambda$ denote our probability space. Hence, an elementary event is an ordered pair (S, λ) where S is an ℓ-bit binary string, and λ a coin model. An experiment consists of the following steps. First select a coin model (by first choosing the number of coins, and then fixing the corresponding biases), next, using this model, generate an alignment sequence (a bit string of length ℓ).

For convenience, we define the following probabilities. For some coin model λ, let $\Pr(\lambda) = \Pr\left(\bigcup_{t \in \{0,1\}^\ell}(t, \lambda)\right)$. Similarly, for $S \in \{0,1\}^\ell$, let $\Pr(S) = \Pr\left(\bigcup_{\mu \in \Lambda}(S, \mu)\right)$. Note that $\sum_{S \in \{0,1\}^\ell} \Pr(S) = \sum_{\lambda \in \Lambda} \Pr(\lambda) = 1$.

Let W_i denote the event that an i-coin model was chosen, i.e., $W_i = \Omega \cap (\{0,1\}^\ell \times \Lambda_i)$. We are interested in the quantities $\Pr(W_i|O) = \Pr(W_i| \bigcup_{\lambda \in \Lambda}(O, \lambda))$ for all $i \geq 1$. Obviously,

$$\Pr(W_i|O) = \frac{\Pr(W_i \wedge O)}{\Pr(O)} = \frac{\Pr(O|W_i) \cdot \Pr(W_i)}{\Pr(O)},$$

where, as above, $\Pr(O|W_i) = \Pr(\bigcup_{\lambda \in \Lambda}(O, \lambda)|W_i)$ and $\Pr(W_i \wedge O) = \Pr(W_i \cap \bigcup_{\lambda \in \Lambda}(O, \lambda))$. Without any prior information, we assume that $\Pr(W_i) = \Pr(W_j)$ for all $i, j \geq 1$. Hence, we need to compute and compare all $\Pr(O|W_i)$ in order to compute the most probable number of coins to generate the sequence.

Single-coin model. A single-coin model λ is defined by p, the probability that a 1 is generated. For discrete valued p,

$$Pr(O \mid W_1) = \sum_{\forall p} Pr(O \mid p \wedge W_1) \times Pr(p \mid W_1).$$

For continuous valued p for which $Pr(p \mid W_1)$ is uniform,

$$\begin{aligned} Pr(O \mid W_1) &= \int_{p=0}^{1} Pr(O \mid p \wedge W_1)\ dp = \int_{p=0}^{1} Pr(O[1] \cdot O[2] \ldots O[\ell] \mid p, W_1)\ dp \\ &= \int_{p=0}^{1} p^k (1-p)^{\ell-k}\ dp = \frac{1}{k+1} - \frac{1}{\ell+1} \end{aligned}$$

where k is the number of 1's in O. Hence, in a single-coin model, provided that the number of 0's and 1's is fixed, their specific locations have no effect on $Pr(O \mid W_1)$, and thus on the likelihood of W_1 given O.

Multiple coin models. For any n-coin model λ and for any sequence O,

$$Pr(O \mid \lambda \wedge W_n) = \sum_{\forall q_1 \ldots q_\ell;\ k \text{ s.t. } q_1 = C_k} \pi_k \cdot p_{q_1}(O[1]) \cdot a_{q_1,q_2} \cdot p_{q_2}(O[2]) \ldots a_{q_{\ell-1},q_\ell} \cdot p_{q_\ell}(O[\ell]),$$

which can be computed using the following recurrence relationship.

Let $\alpha_t(i) = Pr(\text{the coin model } \lambda \text{ generates } O[1:t] \wedge q_t = C_i \mid \lambda \wedge W_n)$. Now,

$$\begin{aligned} \alpha_1(i) &= \pi_i \cdot p_i(O[1]) \quad \text{for all } i, \text{ and} \\ \alpha_{t+1}(j) &= [\sum_{\forall i} \alpha_t(i) \cdot a_{i,j}] \cdot p_j(O[t+1]). \end{aligned}$$

One can simply write

$$Pr(O \mid \lambda \wedge W_n) = \sum_{\forall i} \alpha_\ell(i).$$

To give an example, in a two-coin model $S = \{C_1, C_2\}$. Let $A = \{a_{1,2} = u, a_{1,1} = 1-u, a_{2,1} = v, a_{2,2} = 1-v\}$ and let $B = \{p_1(0) = r, \ \ p_1(1) = 1-r, \ \ p_2(0) = s, \ \ p_2(1) = 1-s\}$. Without any prior information, $\pi = \{\pi_1 = 1/2, \ \ \pi_2 = 1/2\}$. Thus

$$\begin{aligned} Pr(O \mid \lambda \wedge W_2) &= \alpha_\ell(1) + \alpha_\ell(2) \\ &= [\alpha_{\ell-1}(1) \cdot (1-u) + \alpha_{\ell-1}(2) \cdot v] \cdot (1-r)^{O[\ell]} \cdot r^{1-O[\ell]} \\ &\quad + \ [\alpha_{\ell-1}(1) \cdot u + \alpha_{\ell-1}(2) \cdot (1-v)] \cdot (1-s)^{O[\ell]} \cdot s^{1-O[\ell]}. \end{aligned}$$

Iteratively, we can replace the terms involving α_i with those involving α_{i-1}, finally replacing terms involving α_1 with the above definition of α_1, to obtain a multi-variate polynomial on u, v, r, s of total degree 2ℓ with $\frac{1}{4}(\ell^2 - \ell)^2 \leq \frac{\ell^4}{4}$ terms as follows.

Let

$$V_0 = \begin{bmatrix} (1-u) \cdot r & u \cdot s \\ v \cdot r & (1-v) \cdot s \end{bmatrix}$$

and

$$V_1 = \begin{bmatrix} (1-u) \cdot (1-r) & u \cdot (1-s) \\ v \cdot (1-r) & (1-v) \cdot (1-s) \end{bmatrix}.$$

Then one can simply write

$$Pr(O \mid u, v, r, s, W_2) = \begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \cdot \{\prod_{i=1}^{\ell} V_{O[i]}\} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

which can be evaluated by successive multiplications in $O(\sum_{i=1}^{\ell} i^4) = O(\ell^5)$ time. Thus for uniform $Pr(\lambda \mid W_2)$,

$$\begin{aligned} Pr(O \mid W_2) &= \int_{v,u,r,s=0}^{1} Pr(O \mid v, u, r, s, W_2) \ dv \ dr \ du \ ds \\ &= \int_{v,u,r,s=0}^{1} \sum_{\forall i} \alpha_\ell(i) \ dv \ dr \ du \ ds \end{aligned}$$

and thus

$$Pr(O \mid W_2) = \int_{v,u,r,s=0}^{1} \begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \cdot \{\prod_{i=1}^{\ell} V_{O[i]}\} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \; dv \; dr \; du \; ds$$

Notice that one can impose bounds on coin transition probabilities by simply changing the range of integration. It is quite straightforward to determine and perform the symbolic integration of the above multivariate polynomial for $Pr(O|W_2)$ which involves $O(\ell^4)$ terms in $O(\ell^5)$ time.

We now show how to conclude whether the single coin explanation is the likeliest.

Lemma 1 *Knowledge of $Pr(O \mid W_1)$ and $Pr(O \mid W_2)$ is sufficient to conclude whether a single-coin model W_1 has the highest a posteriori probability among all W_i for creating the binary sequence O.*

Proof: An important observation about $Pr(O \mid W_i)$ as a function of i is that it has at most one local maximum; hence for any i-coin model, if $Pr(O \mid W_i) \geq Pr(O \mid W_{i-1})$ and $Pr(O \mid W_i) \geq Pr(O \mid W_{i-2})$, then $Pr(O \mid W_{i-1}) \geq Pr(O \mid W_{i-2})$. Thus if $Pr(O \mid W_1) \geq Pr(O \mid W_2)$, then for any i $Pr(O \mid W_1) \geq Pr(O \mid W_i)$, and thus W_1 is the most likely model for generating O. ∎

The above lemma implies that our test needs to compute and compare $Pr(O \mid W_1)$ and $Pr(O \mid W_2)$, which can be performed in $O(\ell^5)$ time. In the next section we show how to improve this running time to $O(\ell^4 \cdot \log \ell)$.

3.2 Improving the running time

The running time of our test is dominated by the time spent on computing $P(O \mid W_2)$. The multivariate polynomial evaluation in this step can be performed faster than $O(\ell^5)$ time via a divide and conquer approach: The multiplication of two k-variate polynomials where the degree of each variable in a term is bounded above by i can be done in $O(k \cdot i^k \cdot \log i)$ time using FFT. It is not difficult to see that the running time of the divide and conquer algorithm is dominated by that of the final step, which requires multiplying two 2×2 matrices where each entry is a 4-variate polynomial and the degree of each variable is at most ℓ. This leads to an overall running time of $O(\ell^4 \cdot \log \ell)$ for our test, which we state in the theorem below.

Theorem 2 *Given an alignment sequence O of length ℓ, it is possible to determine in $O(\ell^4 \log \ell)$ time whether a posteriori probability that O has been generated by a 1-coin model is higher than that for any other k-coin model for $k > 1$.*

3.3 Examples

For $O = 1\ 1$ (or $O = 0\ 0$)

$$Pr(O \mid W_1) = \int_0^1 (1-r)^2 \; dr = \frac{8}{24}, \quad \textit{and}$$

$$Pr(O \mid W_2) = \int_0^1 \frac{1}{2}[(1-r)^2(1-u) + (1-r)u(1-s) + (1-u)(1-s)^2 + (1-r)v(1-s)] \; dr \; du \; dv \; ds$$

$$= \frac{1}{2} \cdot (\frac{1}{3} \cdot \frac{1}{2} + \frac{1}{2}^3) \cdot 2 = \frac{7}{24}$$

thus it is more likely that O has been generated by a single coin model. This is quite intuitive as a very likely model for generating this sequence consists of a single coin with high bias.
For $O = 1\ 0$ (or $O = 0\ 1$)

$$Pr(O \mid W_1) = \int_0^1 r(1-r)\ dr = \int_0^1 r - r^2\ dr = \frac{4}{24}, \text{ and}$$

$$Pr(O \mid W_2) = \int_0^1 \frac{1}{2}[ru(1-r) + r(1-u)(1-s) + s(1-u)(1-r)^2 + sv(1-s)]\ dr\ du\ dv\ ds$$

$$= \frac{1}{2} \cdot (\frac{1}{6} \cdot \frac{1}{2} + \frac{1}{2}^3) \cdot 2 = \frac{5}{24}$$

thus it is more likely that O has been generated by a two-coin model. This is also intuitive as such a sequence can only be a result of a single coin which is not very biased; however one can think of both biased and unbiased two-coin models that could be responsible of its generation.

We programmed our algorithm to test the likelihood of $Pr(O \mid W_1)$ and $Pr(O \mid W_2)$ on a number of alignment sequences O.

The first table below provides some intuition on the likelihood of models on short sequences. It is interesting to note that the last sequence is much more likely to be generated by a two-coin model due to its periodic nature. The most likely model to generate this sequence would involve two coins which are highly and oppositely biased; the transition probabilities from one coin to the other should also be very high.

O	$Pr(O \mid W_1)$	$Pr(O \mid W_2)$	Likely model
101	0.0833	0.104	W_2
11100	0.0166	0.0208	W_2
111111	0.142	0.0822	W_1
1110111	0.0178	0.0156	W_1
1010101010	0.000360	0.00149	W_2

Here are some sequences which were generated with two coins of opposite biases switched exactly in the middle of each sequence. The test was able to successfully identify bias differences of 10% or more.

O	% *1's in* 1^{st} *half*	% 1's in 2^{nd} half	$Pr(O \mid W_1)$	$Pr(O \mid W_2)$	Likely model
*11101111111111*0000001000000	93%	8%	$1.780 \cdot 10^{-9}$	$2.980 \cdot 10^{-8}$	W_2
*11010110111111*000010101000	77%	25%	$7.396 \cdot 10^{-9}$	$1.117 \cdot 10^{-8}$	W_2
*01001011011*00100101110	55%	45%	$6.163 \cdot 10^{-8}$	$8.450 \cdot 10^{-8}$	W_2

3.4 Extensions

It is possible to extend our statistical test by using a slightly larger alphabet $\{0, 1, -\}$ rather than the binary, where the character "$-$" represents a gap in only one of the sequences in the alignment. This increases the complexity of the problem as two new variables, r' and s', for representing the probabilities of generating a gap for each coin need to be incorporated into the algorithm. The corresponding increase in the number of variables in the multivariate polynomial from 4 to 6 leads to an $O(\ell^6 \log \ell)$ running time.

We also note that an alternative test, which compares $Pr(O \mid W_1, \lambda_1)$ and $Pr(O \mid W_2, \lambda_2)$, where λ_1 and λ_2 are the most likely one-coin and two-coin models respectively can be of use. It is easy to verify that obtaining λ_1 and λ_2 requires a differentiation of the respective univariate and multivariate polynomials and evaluating them at local maxima. This can be done in $O(\ell)$ time for the univariate polynomial, and in $O(\ell^4 \log \ell)$ time for the 4-variate polynomial.

3.5 Identifying all duplicates of a pattern in a long sequence

Given a long sequence S and a pattern Q, it is possible to extend our test to find all segments R of S for which the alignment (Q', R') obtained by an alignment algorithm of choice satisfies (1) $h(Q', R') \leq \delta$ for some threshold value $0 \leq \delta \leq 1$, and (2) the alignment sequence $al(Q', R')$ passes our statistical test. This generalizes available pattern matching algorithms for identifying segments of S that satisfies condition (1) only (some of the better known results in this direction include [12, 19, 8, 16, 7]). A simple implementation which slides Q through S takes $O(|S| \cdot |Q|^4 \log |Q|)$ time.

4 Open problems and discussion

An immediate open problem is whether it is possible to improve the running time of the pattern identification algorithm described above to $O(|S| \cdot |Q|^3 \log |Q|)$ for certain alignments. This raises the issue of generalizing our test, which considers a single alignment between a pair of sequences, to one which considers multiple possible alignments. Another important problem is how to apply this test to "discover" all repeats in a long genome segment, extending the work on sequence discovery algorithms available for non-tandem repeats [1], and other motifs [5, 6, 15] under conventional measures of sequence similarity. One particularly interesting testbed is the identification of the exact boundaries of multi-layered tandemly repeated DNA segments. A practical approach to this problem is to slide a fixed size window across the sequence of interest, measuring the percentage similarity score of every window position w_i with every other w_j. It is expected that for those w_i and w_j for which $j - i + 1$ is a multiple of a *period* size, the percentage similarity score will be higher than other window positions; thus one can view each w_i, w_j pair whose similarity score is higher than a threshold as evidence that $k = j - i + 1$ is a candidate period size (usually on a 2-D plot). If the candidate period size k is supported by sufficient evidence, one may conclude that k is indeed the size of a period. Although this approach has been used in a number of applications, it raises a few issues.

(1) The widely accepted hypothesis for high order tandem repeat evolution (e.g. the high repeat alpha-satellite DNA) maintains that some early tandem duplications at the monomeric level are

followed by a k-mer duplication event, after which almost all duplication events occur at k-meric level [17, 13]. In other words duplication events occur hierarchically in time, and "larger period" sizes are always multiples of "smaller period" sizes.
However, one can imagine duplications occurring in a number of different block sizes scattered over the sequence; this may lead the above strategy to fail to correctly identify the high order in the repeat pattern.
(2) Different window sizes may lead to different conclusions.
(i) if the window size is smaller than the size of a period, the method will not compare full periods against each other and the results derived can be misleading;
(ii) if the window size is much larger than the size of a period, then the variations in similarity between w_i, w_j pairs will be insignificant.
(3) The thresholds for (i) the similarity score and (ii) the number of evidences for identifying a potential period as an actual period play a significant role in the method. If the threshold values are too small, there will be too many periods to report; if they are too large, some of the periods may be ignored.

References

[1] E. F. Adebiyi, T. Jiang, M. Kaufmann, An Efficient Algorithm for Finding Short Approximate Non-Tandem Repeats, *In Proceedings of ISMB 2001.*

[2] A. N. Arslan, O. Egecioglu, P. A. Pevzner A new approach to sequence comparison: normalized sequence alignment, *Proceedings of RECOMB 2001.*

[3] Bailey JA, Yavor AM, Massa HF, Trask BJ, Eichler EE. Segmental duplications: organization and impact within the current human genome project assembly, *Genome Research* 11(6), Jun 2001.

[4] T. Bailey, C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers, *Proceedings of ISMB* 1994, AAAI Press.

[5] J. Buhler and M. Tompa Finding Motifs Using Random Projections, *In Proc. of RECOMB 2001.*

[6] J. Buhler Efficient Large Scale Sequence Comparison by Locality Sensitive Hashing, *Bioinformatics*17(5), 2001.

[7] Richard Cole and Ramesh Hariharan, Approximate String Matching: A Simpler Faster Algorithm, *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pp. 463-472, 25-27 January 1998.

[8] W. Chang and E. Lawler, Approximate String Matching in Sublinear Expected Time, *Proc. IEEE Symposium on Foundations of Computer Science*, 1990.

[9] J. E. Horvath, L. Viggiano, B. J. Loftus, M. D. Adams, N. Archidiacono, M. Rocchi, E. E. Eichler Molecular structure and evolution of an alpha satellite/non-satellite junction at 16p11. *Human Molecular Genetics*, 2000, Vol 9, No 1.

[10] Jackson, Strachan, Dover, *Human Genome Evolution*, *Bios Scientific Publishers*, 1996.

[11] E. S. Lander et al., Initial sequencing and analysis of the human genome, *Nature*, 15:409, Feb 2001.

[12] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, *Cybernetics and Control Theory*, 10(8):707-710, 1966.

[13] T. Mashkova, N. Oparina, I. Alexandrov, O. Zinovieva, A. Marusina, Y. Yurov, M. Lacroix, L. Kisselev, Unequal crossover is involved in human alpha satellite DNA rearrangements on a border of the satellite domain, *FEBS Letters*, 441 (1998).

[14] A. Marzal and E. Vidal, Computation of normalized edit distances and applications, *IEEE Trans. on PAMI*, 15(9):926-932, 1993.

[15] L. Parida, I. Rigoutsos, A. Floratsas, D. Platt, Y. Gao, Pattern discovery on character sets and real valued data: linear bound on irredundant motifs and an efficient polynomial time algorithm, *Proceedings of ACM-SIAM SODA, 2000.*

[16] S. C. Sahinalp and U. Vishkin, Approximate and Dynamic Matching of Patterns Using a Labeling Paradigm, *Proc. IEEE Symposium on Foundations of Computer Science*, 1996.

[17] George P. Smith Evolution of Repeated DNA Sequences by Unequal Crossover, *Science*, vol 191, pp 528–535.

[18] J. D. Thompson, D. G. Higgins, T. J. Gibson, Clustal-W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice, *Nucleic Acid Research* 1994, Vol. 22, No. 22.

[19] E. Ukkonen, On Approximate String Matching, *Proc. Conference on Foundations of Computation Theory*, 1983.

[20] C. Venter et. al., The sequence of the human genome, *Science*, 16:291, Feb 2001.