Journal of Bioinformatics and Computational Biology
Vol. 2, No. 4 (2004) 657–668
© Imperial College Press

Imperial College Press
www.icpress.co.uk

IDENTIFYING UNIFORMLY MUTATED SEGMENTS WITHIN REPEATS

S. CENK SAHINALP[1], EVAN EICHLER[2], PAUL GOLDBERG[3],
PETRA BERENBRINK[4], TOM FRIEDETZKY[5] and FUNDA ERGUN[6]

[1]*School of Computing Science, Simon Fraser University, Canada*
cenk@cs.sfu.ca

[2]*Department of Genetics and Center for Computational Genomics, CWRU, USA*
eee@po.cwru.edu

[3]*Department of Computer Science, University of Warwick, UK*
pwg@dcs.warwick.ac.uk

[4]*School of Computing, Simon Fraser University, Canada*
petra@cs.sfu.ca

[5]*Pacific Institute of Mathematics, Simon Fraser University, Canada*
tf@pims.math.ca

[6]*School of Computing, Simon Fraser University, Canada*
funda@cs.sfu.ca

Received 28 April 2003
Revised 22 March 2004
Accepted 4 April 2004

Given a long string of characters from a constant size 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 n-coin models for generating a binary string S, where each bit of S is generated via an independent toss of one of the n 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 procedure to evaluate the likelihood of a n-coin model for given S, subject a uniform prior distribution over the parameters of the model (that represent mutation rates and probabilities of copying events). In the absence of detailed prior knowledge of these parameters, the algorithm can be used to determine whether the *a posteriori* probability for $n = 1$ is higher than for any other $n > 1$. 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 assumed convexity of the *a posteriori* probability for n.

Our test can be used in the analysis of long alignments between pairs of genomic sequences in a number of ways. For example, functional regions in genome sequences exhibit much lower mutation rates than non-functional regions. Because our test provides means for determining variations in the mutation rate, it may be used to distinguish functional regions from non-functional ones. Another application is in determining whether two highly similar, thus evolutionarily related, genome segments are the result of a single copy event or of a complex series of copy events. This is particularly an issue in evolutionary studies of genome regions rich with repeat segments (especially tandemly repeated segments).

Keywords:

1. Introduction

The human genome consists of numerous segments "repeated" with various degrees of similarity.[3,17,27] Long repeat sequences are more likely to be generated as a result of segmental copies during evolution rather than by chance.[a] Approximately 60% of the human genome appears to be repeated.[b]

Repeat segments are commonly classified into three main categories. Over 45% of the human genome comprises common repeats; one example is the ∼300 bp *alu* element, occurring more than 1M times within a divergence rate of 5%−15%. Another ∼5% consists of the centromeric repeats, particularly the alpha satellite and microsatellite DNA. A final ∼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 copying followed by point mutations provide the main process underlying genome evolution.[16]

Several biochemical mechanisms underlying segmental copies have been identified in the last 30 years (e.g. *unequal cross over*,[23] *replication slippage* and *retrotransposition*); potentially many more are waiting to be discovered. The task of identifying all copying 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 copy event during evolution. Note that a *repeat* can be a result of multiple complex *copy* events: for example a long tandemly repeated sequence S, S, S, S may be a result of tandemly copying the first S three times or copying the first S once to obtain S, S and copying this whole segment again to get S, S, S, S; there are many other possibilities. We address the problem of identifying these copying events and the order in which they occurred.

1.1. *Contributions*

In this paper we present a probabilistic test for identifying whether a pair of genome sequences with a high similarity score are indeed a result of a single copy 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 × 10^{-9} per base pair per year for non-functional segments of humanoids and old world monkeys).

[a] Given a sequence of size 3×10^9 bp (the size of the human genome) generated by an i.i.d. random source on the four letter DNA alphabet, the probability of having a pair of 100 bp segments with Hamming distance of 5 or less is smaller than 2^{-75} (practically nil).

[b] The chance that an arbitrary 1 Kbp segment in the human genome to have a corresponding segment with 30% divergence is 60%.

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 copy event. Because after a copying event 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 copies between two genome regions is to iteratively locate 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 Arslan *et al.*,[2] where an alternative "normalized" similarity measure based on Marzal and Vidal's work[20] 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 copy 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 copying events, including their order, as well as the source and destination subsequences.[15] Assuring that the sequences have been subject to independent point mutations only, rather than a complex series of copying events, is critical to the accuracy of phylogenetic analysis, especially based on distance comparisons (e.g. Thompson *et al.*[24]) involving these sequences.

To address the above issue we propose a new method for pairwise sequence comparison in the form of a *probabilistic test* to determine whether a given pair of sequences with high similarity score have been generated as the result of a single copy event. More specifically, we consider n-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 n-state Hidden Markov Models, those for which $n = 1$ are more likely than any other for which $n > 1$ (we compare the aggregate likelihood of all 1-state HMMs with that of n-state HMMs for $n > 1$). 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 n.

Similar problems have been considered earlier in Refs. 9, 11, 14, 26. In fact Ref. 9 considered a two state HMM for identifying the cutoff point between one mutation rate and another for a *given* alignment sequence S. The *most likely* HMM is constructed through standard expectation maximization (EM) techniques. In contrast we focus on the *aggregate* effects of all possible 1 or 2 state HMMs rather than focusing on a single model for robustness purposes as it does not require specification of a cut-off point for differentiating one coin and two coin models. Furthermore our approach need not consider a single alignment between the pair of sequences considered: it is possible to generalize our method to aggregate over all possible alignments according to the likelihood of their occurrence.

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 R and S denote genome segments, $R[t]$ denotes the character in position t of segment R, and $R[t : u]$ the substring between the characters in positions t and u (inclusive) of R. $|R|$ denotes the length of the segment R.

An alignment between two genome segments R and S is a pair (R', S'), where $R', S' \in \{a, c, g, t, -\}^\ell$ for some $\ell = |R'| = |S'|$, such that R and S are obtained if all "$-$" are removed from R' and S' respectively. Furthermore, there should be no t such that $R'[t] = S'[t] = -$.

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 (R', S'), where $|R'| = |S'| = \ell$, let $al(R', S')$, denote the *alignment sequence* of (R', S') whose position t is $R'[t] \oplus S'[t]$. We denote by $h(R', S')$ the normalized Hamming distance between R' and S', i.e. the number of 1's in $al(R', S')$ divided by ℓ.

3. A Probabilistic Test for Detecting Simple Copies

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

The underlying motivation for the above problem is the need to test whether the sequences R and S have been generated by a single copying event, followed by independent point mutations only. Alternatively they could either be a result of more complex sequence of multiple copying events, where the segments involved were subjected to mutations for different periods of time, or a result of a single copying 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 R' and S'. Thus a probabilistic test for determining whether the edit errors between R' 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 copying 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 $T = al(R', S') = (T[1], \ldots, T[\ell])$ of length ℓ, we would like to compute the *a posteriori* probability that T 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 a 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_t, the coin used in generating $T[t]$, $1 \leq t \leq \ell$. Note that for any $t = 1, \ldots, \ell$, $\Pr(T[t] = 1 \mid q_t = C_j) = p_j(1)$, since the outcome does not depend on the location itself but only on the "active" coin. Furthermore, let $a_{i,j}$ denote the transition probability $\Pr(q_{t+1} = C_j \mid q_t = C_i)$ that coin C_i is replaced by C_j between locations t and $t + 1$. Note that $a_{i,j}$ does not depend on the location t. 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).

Letting A be the $n \times n$ matrix with $A(i, j) = a_{i,j}$, P be the n-dimensional vector with $P(j) = p_j(1)$ and π be the n-dimensional vector with $\pi(j) = \pi_j$, an n-coin model λ is now defined by the triple (A, P, π). For $n \geq 1$, let Λ_n denote the set of all n-coin models. Denote by $\Lambda = \bigcup_{n \geq 1} \Lambda_n$ 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 (B, λ) where B 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 n, then fixing the parameters (A, P, π)), next, use this model to generate an alignment sequence B (a bit string of length ℓ).

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

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 $\Pr(W_i \mid T) = \Pr\big(W_i \mid \bigcup_{\lambda\in\Lambda}(T,\lambda)\big)$ for all $i \geq 1$. By Bayes' rule:

$$\Pr(W_i \mid T) = \frac{\Pr(W_i \wedge T)}{\Pr(T)} = \frac{\Pr(T \mid W_i)\cdot\Pr(W_i)}{\Pr(T)},$$

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

Single-coin model. Let λ_p denote the single-coin model where p is the probability that a 1 is generated by the coin. For discrete valued p,

$$\Pr(T \mid W_1) = \sum_p \Pr(T \mid \lambda_p \wedge W_1) \times \Pr(\lambda_p \mid W_1).$$

For continuous valued p for which $\Pr(\lambda_p \mid W_1)$ is uniform over $p \in [0,1]$,

$$\begin{aligned}\Pr(T \mid W_1) &= \int_0^1 \Pr(T \mid \lambda_p \wedge W_1)dp = \int_0^1 \Pr(T[1]\cdot T[2]\cdots T[\ell] \mid \lambda_p, W_1)dp\\ &= \int_0^1 p^k(1-p)^{\ell-k}dp\\ &= \sum_{i=0}^{k+1}\binom{\ell-k}{i}\frac{1}{\ell-i+1},\end{aligned}$$

where k is the number of 1's in T. Hence, in a single-coin model, provided that the number of 0's and 1's is given, their specific locations have no effect on $\Pr(T \mid W_1)$, and thus on the likelihood of W_1 given T.

Multiple coin models. For coins q_i, q_{i+1} let $a_{q_i,q_{i+1}}$ be the transition probability for moving from q_i to q_{i+1}; let $p_{q_i}(b)$ be the probability that coin q_i outputs b (for $b = 0$ or 1), and let π_{q_1} denote the probability π_k for k such that $q_1 = C_k$. Then for any n-coin model λ and for any sequence T,

$$\Pr(T \mid \lambda) = \sum_{q_1\cdots q_\ell} \pi_{q_1}\cdot p_{q_1}(T[1])\cdot a_{q_1,q_2}\cdot p_{q_2}(T[2])\cdots a_{q_{\ell-1},q_\ell}\cdot p_{q_\ell}(T[\ell]),$$

which can be computed by dynamic programming using the following recurrence relationship.

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

$$\alpha_1(i) = \pi_i \cdot p_i(T[1]) \quad \text{for all } i, \quad \text{and}$$
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i) \cdot a_{i,j}\right] \cdot p_j(T[t+1]).$$

One can then write

$$\Pr''(T \mid \lambda \wedge W_n) = \sum_{i=1}^{n} \alpha_\ell(i).$$

To give an example, if λ is 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 $P = \{p_1(0) = r,\ p_1(1) = 1-r,\ p_2(0) = s, p_2(1) = 1-s\}$. Under a non-informative prior, $\pi = \{\pi_1 = 1/2,\ \pi_2 = 1/2\}$. Thus

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

Iteratively, we can express the terms involving $\alpha_i(t)$ in terms of those involving $\alpha_{i-1}(t)$, 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

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

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

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

and thus

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

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 $\Pr(T \mid W_2)$ 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.

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

The assumption is derived from an observation in Bishop[5] (Chapter 10) regarding model order selection in the Bayesian setting; that the likelihood of the data as a function of model complexity typically increases to a peak and then decreases monotonically. For our purposes this suggests that $\Pr(T \mid W_n)$ as a function of n has at most one local maximum; hence for any n-coin model, if $\Pr(T \mid W_n) \geq \Pr(T \mid W_{n+1})$ then $\Pr(T \mid W_{n+1}) \geq \Pr(T \mid W_{n+2})$. Thus if $\Pr(T \mid W_1) \geq \Pr(T \mid W_2)$, then for any n $\Pr(T \mid W_1) \geq \Pr(T \mid W_n)$, and thus W_1 is the most likely model for generating T.

The above assumption implies that our test needs to compute and compare $\Pr(T \mid W_1)$ and $\Pr(T \mid W_2)$, which can be performed in $O(\ell^5)$ time. We notice that 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 corollary below.

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

3.2. *Examples*

For $T = 1\ 1$ (or $T = 0\ 0$),

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

$$\begin{aligned}\Pr(T \mid W_2) &= \int_0^1 \frac{1}{2}[(1-r)^2(1-u) + (1-r)u(1-s) + (1-u)(1-s)^2 \\ &\quad + (1-r)v(1-s)] dr\, du\, dv\, ds \\ &= \frac{1}{2} \cdot \left(\frac{1}{3} \cdot \frac{1}{2} + \frac{1}{2}^3\right) \cdot 2 = \frac{7}{24},\end{aligned}$$

thus it is more likely that T 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 $T = 1\ 0$ (or $T = 0\ 1$),

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

$$\begin{aligned}\Pr(T \mid W_2) &= \int_0^1 \frac{1}{2}[ru(1-r) + r(1-u)(1-s) + s(1-u)(1-r)^2 \\ &\quad + sv(1-s)]dr\,du\,dv\,ds \\ &= \frac{1}{2} \cdot \left(\frac{1}{6} \cdot \frac{1}{2} + \frac{1}{2}^3\right) \cdot 2 = \frac{5}{24},\end{aligned}$$

thus it is more likely that T 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 $\Pr(T \mid W_1)$ and $\Pr(T \mid W_2)$ on a number of alignment sequences T.

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.

T	$\Pr(T \mid W_1)$	$\Pr(T \mid W_2)$	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.

T	% 1's in 1st Half	% 1's in 2nd Half	$\Pr(T \mid W_1)$	$\Pr(T \mid W_2)$	Likely Model
*11101111111111*0000001000000	93	7	$1.780 \cdot 10^{-9}$	$2.980 \cdot 10^{-8}$	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

3.3. *Extensions*

It is possible to extend our probabilistic 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 $\Pr(T \mid W_1, \lambda_1)$ and $\Pr(T \mid W_2, \lambda_2)$, 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.

3.4. *Identifying all copies 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 probabilistic test. This generalizes available pattern matching algorithms for identifying segments of S that satisfies condition (1) only (the literature contains some of the better known results in this direction[8,10,18,22,25]). 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[6,7,21] 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 copies at the monomeric level are followed by a k-mer copying event, after which almost all copying events occur at k-meric level.[19,23] In other words copying events occur hierarchically in time, and "larger period" sizes are always multiples of "smaller period" sizes.

However, one can imagine copies 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. Adebiyi EF, Jiang T, Kaufmann M, An efficient algorithm for finding short approximate non-tandem repeats, *Proceedings of ISMB* 2001.
2. Arslan AN, Egecioglu O, Pevzner PA, 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), 2001.
4. Bailey T, Elkan C, Fitting a mixture model by expectation maximization to discover motifs in biopolymers, *Proceedings of ISMB*, AAAI Press, 1994.
5. Bishop CM, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
6. Buhler J, Tompa M, Finding motifs using random projections, *Proc. of RECOMB* 2001.
7. Buhler J, Efficient large scale sequence comparison by locality sensitive hashing, *Bioinformatics* **17**(5), 2001.
8. Richard Cole, Ramesh Hariharan, Approximate string matching: a simpler faster algorithm, *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pp. 25–27, 463–472, 1998.
9. Churchill GA, Stochastic models for heterogeneous DNA sequences, *Bulletin of Mathemathical Biology* **51**:79–94, 1989.
10. Chang W, Lawler E, Approximate string matching in sublinear expected time, *Proc. IEEE Symposium on Foundations of Computer Science*, 1990.

11. Fu Y-X, Curnow RN, Maximum likelihood estimation of multiple change points, *Biometrika* **77**:563–573, 1990.
12. Green PJ, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika* **82**:711–732, 1995.
13. Halpern AL, Minimally selected p and other tests for a single abrupt changepoint in a binary sequence, *Biometrics* **55**, 1999.
14. Halpern AL, Multiple changepoint testing for an alternating segments model of a binary sequence, *Biometrics* **56**, 2000.
15. Horvath JE, Viggiano L, Loftus BJ, Adams MD, Archidiacono N, Rocchi M, Eichler EE, Molecular structure and evolution of an alpha satellite/non-satellite junction at 16p11. *Hum Mol Gent* **9**(1), 2000.
16. Jackson, Strachan, Dover, Human genome evolution, *Bios Scientific Publishers*, 1996.
17. Lander ES *et al.*, Initial sequencing and analysis of the human genome, *Nature* **15**:409, 2001.
18. Levenshtein VI, Binary codes capable of correcting deletions, insertions and reversals, *Cybernet Control Theor* **10**(8):707–710, 1966.
19. Mashkova T, Oparina N, Alexandrov I, Zinovieva O, Marusina A, Yurov Y, Lacroix M, Kisselev L, Unequal crossover is involved in human alpha satellite DNA rearrangements on a border of the satellite domain, *FEBS Lett* **441**, 1998.
20. Marzal A, Vidal E, Computation of normalized edit distances and applications, *IEEE T PAMI* **15**(9):926–932, 1993.
21. Parida L, Rigoutsos I, Floratsas A, Platt D, Gao Y, 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.
22. Sahinalp SC, Vishkin U, Approximate and dynamic matching of patterns using a labeling paradigm, *Proc. IEEE Symposium on Foundations of Computer Science*, 1996.
23. George P. Smith, Evolution of repeated DNA sequences by unequal crossover, *Science* **191**:528–535.
24. Thompson JD, Higgins DG, Gibson TJ, Clustal-W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice, *Nucleic Acid Res* **22**(22), 1994.
25. Ukkonen E, On approximate string matching, *Proc. Conference on Foundations of Computation Theory*, 1983.
26. Venter J, Steel S, Finding multiple abrupt change points, *Comput Stat Data Anal* **22**:481–501, 1996.
27. Venter C *et al.*, The sequence of the human genome, *Science* **16**:291, 2001.