Efficient and exact maximum likelihood quantization of genomic features using dynamic programming

Mingzhou (Joe) Song*

Department of Computer Science,

New Mexico State University,

Las Cruces, NM 88003, USA

Fax: +1 575 646 1002 E-mail: joemsong@cs.nmsu.edu

*Corresponding author

Robert M. Haralick

Ph.D. Program in Computer Science,

Graduate Center, City University of New York,

New York, NY 10016, USA

Fax: +1 212 817 1510 E-mail: haralick@ptah.gc.cuny.edu

Stéphane Boissinot

Department of Biology,

Queens College, City University of New York,

Flushing, NY 11367, USA

Fax: +1 718 997 3445 E-mail: stephane.boissinot@qc.cuny.edu

Abstract: An efficient and exact dynamic programming algorithm is introduced to quantize a continuous random variable into a discrete random variable that maximizes the likelihood of the quantized probability distribution for the original continuous random variable. Quantization

is often useful before statistical analysis and modeling of large discrete network models from observations of multiple continuous random variables. The quantization algorithm is applied to genomic features including the recombination rate distribution across the chromosomes and the non-coding transposable element LINE-1 in the human genome. The association pattern is studied between the recombination rate, obtained by quantization at genomic locations around LINE-1 elements, and the length groups of LINE-1 elements, also obtained by quantization on LINE-1 length. The exact and density-preserving quantization approach provides an alternative superior to the inexact and distance-based univariate iterative k-means clustering algorithm for discretization.

Keywords: Quantization; Discretization, Dynamic Programming; Recombination Rate Distribution; Transposable Elements; LINE-1.

Reference to this paper should be made as follows: Song, M., Haralick R.M., and Boissinot, S. (2008) 'Efficient and exact maximum likelihood quantization of genomic features using dynamic programming', *Int. J. Data Mining and Bioinformatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical Notes:

Mingzhou (Joe) Song received his Ph.D. in 2002 and M.S. in Electrical Engineering in 1999, both from the University of Washington, Seattle. He obtained his B.S. in Electrical Engineering at Beijing University of Posts and Telecommunications in 1992. He has been an Assistant Professor at the Department of Computer Science, New Mexico State University since 2005. He was Assistant Professor of Computer Science with Queens College and the doctoral faculty with Graduate Center, City University of New York from 2002 to 2005. His research areas include data mining, computational modeling, quantitative biology, and computer vision. He has collaborated with life scientists to solve computational modeling problems involved in biofuels, cancer, neuroscience, and microbial communities.

Robert M. Haralick received his B.A. in Mathematics in 1964, B.S. in Electrical Engineering in 1966, M.S. in Electrical Engineering in 1967, and Ph.D. in 1969, all from the University of Kansas. He is a Dis-

tinguished Professor of Computer Science, Graduate Center, City University of New York. He held the Clairmont Egtvedt Professorship in Electrical Engineering with University of Washington. He has made a series of contributions in computer vision and pattern recognition. His most recent interests have been in manifold clustering in high dimensional spaces. He is a Fellow of IEEE and a Fellow of IAPR. He has served as editors for a number of journals in computer vision and pattern recognition.

Stéphane Boissinot received his Ph.D. in 1994 from the University of Montpellier II, France. He was a post-doctoral fellow at the University of Texas, Houston from 1994 to 1996 and at the National Institutes of Health, Bethesda from 1997 to 2002. In 2003, he joined the faculty in the Department of Biology at Queens College, the City University of New York, where he is currently associate professor. He is also faculty at the Graduate Center of the City University of New York. His research interests are the evolution retrotransposons in vertebrate genomes and the evolution of resistance to viral infection.

1 Introduction

Quantization is a monotonically increasing transformation that converts a continuous random variable to a discrete random variable. Quantization functions that better preserve the original probability density function (p.d.f.) legitimize the transfer of statistical analysis and modeling performed on the discrete random variable back to the original continuous random variable. We present an efficient and exact algorithm that achieves such a density-preserving quantization by dynamic programming. The optimality of the discretization is guaranteed by a general mapped additivity satisfied by all major quantization criteria. In our optimal quantization algorithm, the most important regions are finely quantized, while less important regions are coarsely quantized, statistically much more efficient than a uniform quantization. Other methods, e.g., kernel methods, treat everywhere in a

space equally without the prioritized resource allocation. For the less important regions, there is the potential wasting of resources. The algorithm can work on either continuous data points or counts of data already accumulated in finer-than-desired bins. The number of quantization levels is determined by either the Bayesian information criterion – a function of the log likelihood, the sample size, and the number of quantization levels, or cross validation.

Graphical modeling of multiple random variables has motivated continued research on quantization algorithms. A graphical model uses a graph to represent the joint probability distribution function of multiple random variables. Each node in the graph represents a random variable. Edges between nodes encode statistical dependencies among variables. The joint probability distribution function can be decomposed to the product of conditional probability functions of variables at each node given their parent nodes. A graphical model of continuous random variables typically makes parametric assumptions on the conditional probabilities for each node in the graph, but not so for a graphical model of discrete random variables. Thus discretization is often necessary for graphical modeling if no prior knowledge is available on the forms of conditional probabilities for each continuous random variable in question. Additionally, there are more alternatives (Margaritis and Thrun, 2001) to determine statistical independencies between discrete random variables than for continuous ones when the underlying p.d.f. is unknown.

Relevant to our work are approaches that find a quantization of the data by optimizing an objective function. Entropy (Haralick, 1976), likelihood (Hearne and Wegman, 1992), and distance have been used as objective functions. Among these criterions, only likelihood ties directly to the p.d.f. of the original continuous random variable. A less-known optimal solution (Wu, 1992) using dynamic programming has been provided for the univariate k-means problem. Fulton et al. (1995) have later used dynamic programming to find an optimal quantization to classify a univariate sample. However, dynamic programming has not been used in density-preserving quantization. Our methodology obtains a non-uniform quanti-

zation by optimizing an objective function that combines likelihood and entropy. Optimal quantization ensures the adaptivity to the data and overcomes the statistical ineffectiveness of uniform quantization.

We applied our quantization algorithm to genomic features including the recombination rate and the distribution of Long Interspersed Nuclear Element LINE-1 (L1) in the human genome. The association pattern is studied between the recombination rate, obtained by quantization at genomic locations around L1 elements, and the length groups of L1 elements, also obtained by quantization on L1 length.

The paper is organized into seven sections. Following Section 1 the introduction, we define the density-preserving quantization objective function in Section 2; the optimality condition for efficient finding a quantization by dynamic programming is discussed in Section 3; the dynamic programming algorithm for the quantization is designed and analyzed in Section 4; quantization results of the recombination rate distribution function in human genome are presented in Section 5; the association of quantized length groups of L1 with the recombination rate is discovered in Section 6; finally, we draw our conclusions in Section 7.

2 The Likelihood of Quantization

We define and justify a quantization objective function that includes the likelihood and entropy measures on the observed data set. Let X be a continuous random variable with p.d.f. $p(x)$. Let calligraphic $\mathcal{X} = \langle x_1, x_2, \ldots, x_N \rangle$ be a sorted sequence of a random sample of size N from X, where $x_1 \leq x_2 \leq \ldots \leq x_N$. We define $\mathcal{X}_m^n$ as the subsequence $\langle x_m, x_{m+1}, \ldots, x_n \rangle$. Let Q be an L-level quantization with decision boundaries $B = \{b_0, b_1, \ldots, b_L\}$, $b_0 < b_1 < \ldots < b_L$. Let $\Delta(q)$ be the width of bin q. Let N_q be the total number of data points in bin q. Let $\hat{p}(x)$ be the p.d.f. derived from the histogram of the observed data using quantization Q.

To preserve the original p.d.f. $p(x)$, one can minimize the Kullback-Leibler divergence from $\hat{p}(x)$ to $p(x)$, defined as

$$D_{KL}(p||\hat{p}) = \int p(x) \log \frac{p(x)}{\hat{p}(x)} dx = \mathbf{E}[\log p(X)] - \mathbf{E}[\log \hat{p}(X)].$$

As $p(x)$ is fixed, minimizing $D_{KL}(p||\hat{p})$ is equivalent to maximizing $\mathbf{E}[\log \hat{p}(X)]$. Let $\bar{p}_q$ be the estimated average probability density of bin q computed by

$$\bar{p}_q = \frac{N_q/N}{\Delta(q)}. \tag{1}$$

We estimate $\mathbf{E}[\log \hat{p}(X)]$ by the average sample log likelihood. Thus the *log likelihood* of X for quantization Q is

$$J(X|Q) = \mathbf{E}[\log \hat{p}(X)] = \frac{1}{N} \sum_{q=1}^{L} N_q \log \bar{p}_q = \sum_{q=1}^{L} J(X|q), \tag{2}$$

where

$$J(X|q) = \frac{N_q}{N} \log \bar{p}_q$$

is the contribution from the single bin q.

While entropy has been utilized as a class impurity measure (Breiman et al., 1984), we use entropy to characterize the generalization ability of quantization. Maximizing entropy corresponds to minimizing information loss. Entropy is defined by

$$H(X|Q) = -\sum_{q=1}^{L} \frac{N_q}{N} \log \frac{N_q}{N} = \sum_{q=1}^{L} H(X|q), \tag{3}$$

where

$$H(X|q) = \frac{N_q}{N} \log \frac{N}{N_q}$$

is the contribution from the single bin q. Examples of maximum entropy quantization include equal probability quantization (Haralick et al., 1973), histogram equalization (Jain, 1989), Voronoi tessellation (Voronoi, 1908), or more generally, nearest neighbor partitions (Gersho and Gray, 1992).

In contrast to likelihood, entropy is not a direct performance measure of pattern recognition test results. Rather, the entropy measure in our context controls over-fitting. The larger the entropy, the less likely the possibility of over-quantization.

We define the quantization objective function or performance measure as

$$T(X|Q) = w_J J(X|Q) + w_H H(X|Q) \tag{4}$$

with

$$w_J + w_H = 1, w_J, w_H \geq 0,$$

where w_J and w_H are given weights for log likelihood and entropy, respectively. This first term will allow a best fit to the data while the second term prevents over-fitting. If we define $T(X|q)$, the contribution from a single bin q, as

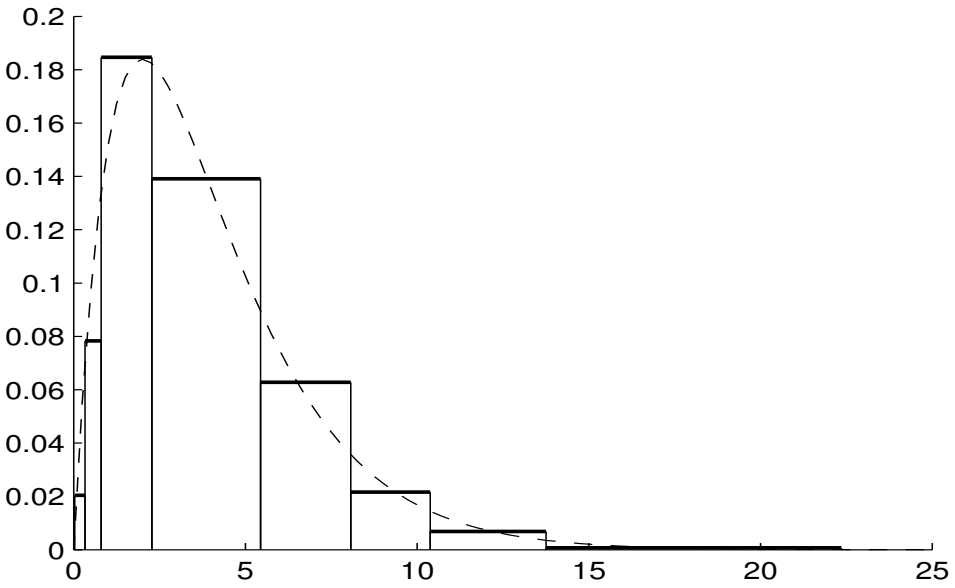
$$T(X|q) = w_J J(X|q) + w_H H(X|q).$$

$T(X|Q)$ can be written in an additive form as

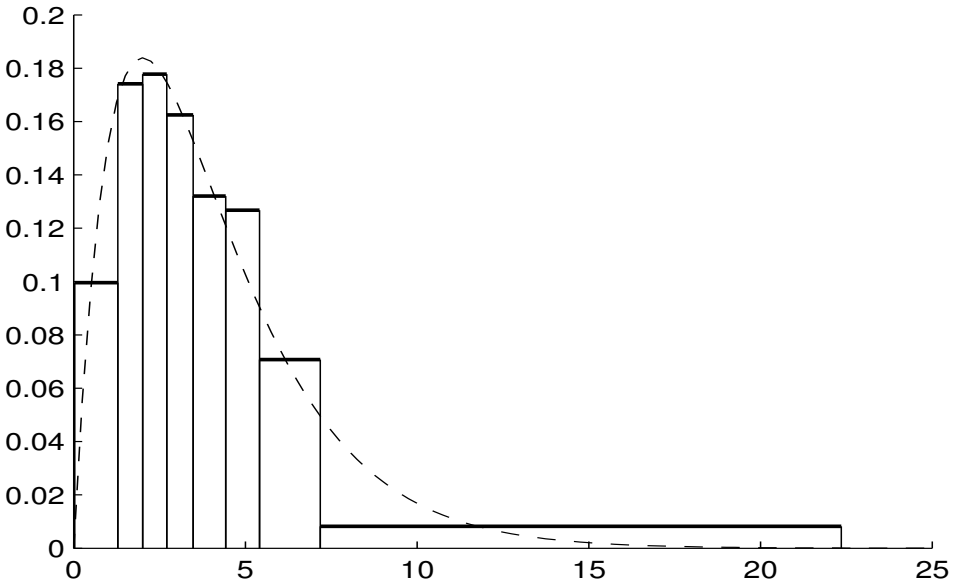
$$T(X|Q) = \sum_{q=1}^{L} T(X|q) \tag{5}$$

A data-driven strategy is to determine the coefficients w_J, w_H through cross validation. The values of w_J, w_H that maximize the likelihood of the left-out fold are selected to be the coefficients. The number of quantization levels is determined by either the Bayesian information criterion – a function of the log likelihood, the sample size, and the number of quantization levels, or cross validation.

Example. We illustrate with a Chi-squared example that contrasts maximum likelihood and maximum entropy quantization. Our example has 1000 data points generated using a Chi-squared distribution with 4 degrees of freedom. The number of quantization levels was 8. The density estimates are shown in Fig. 1. The dashed line is the original Chi-squared p.d.f. In Fig. 1(a), it is evident that the underlying density changes much more rapidly in $[0, 2]$ than in $[2, \infty)$. The bins are narrower for the region from 0 to 2 than for the region above 2, corroborating the consistency result in (Scott, 1992). In Fig. 1(b), the bins for the region around the mode at 2



(a) Maximum likelihood quantization ($w_J = 1, w_H = 0$).



(b) Maximum entropy quantization ($w_J = 0, w_H = 1$).

Figure 1: Density estimates of Chi-squared data using optimal quantization.

are narrower than the region further away from the mode. The density of the region around the mode is larger than other regions. When entropy is maximized, each bin contains about the same number of points. This naturally leads to narrower bins for regions of higher density and wider bins for regions of lower density. The rationale behind the entropy measure is that the least commitment should be made to the sample. This controls the generalization ability of the quantization. On the other hand, the maximum likelihood approach finds the best fit to the data and it may over-fit. Therefore, it is necessary to combine the two measures in a controlled fashion as we have done in defining $T(X|Q)$, which is especially important when the sample size is small.

3 The Optimality Condition for Quantization using Dynamic Programming

Given the sorted data sequence $\mathcal{X}$ and the number of quantization levels L, the goal of quantization is to find an optimal quantizer Q^* such that a pre-defined objective function $T(\mathcal{X}|Q)$ is maximized by Q^*. An efficient solution of such a problem is still open for multivariate random variables. However, an efficient dynamic programming solution exists for optimal quantization of a univariate random variable given that the quantization performance measure satisfies a very general mapped additivity condition.

Definition 3.1. *(Sub-quantizer) Q_r^u is called a sub-quantizer of quantizer Q if it has $u - r + 1$ quantization levels and the decision boundaries are the same with those for intervals from r to u of Q. We define $T(\mathcal{X}_m^n|Q_r^u)$ as the performance measure of the sub-quantization, evaluated on the subsequence $\mathcal{X}_m^n$ of $\mathcal{X}$ that falls in the bins of Q_r^u.*

The performance measure of a sub-quantizer is exactly the contributions from the data points and intervals it covers. Notice that such defined sub-quantizer per-

formance measure may be different from the performance measure of an isolated quantizer that covers just the same points and intervals. For the performance measure defined in Eq. (12) that involves Eqs. (2) and (3), N is still defined on the overall data set $\mathcal{X}_N$ even when computing sub-quantizer performance measures.

Definition 3.2. *(Mapped additivity) The mapped additivity condition is that the mapped performance measure of any quantizer Q on a given data set is additive over mapped performance measures of any combination of sub-quantizers of Q, when there is a monotonically increasing function that can achieve the mapping. Let $g(x)$ be such a monotonically increasing function defined on the domain of $T(X|Q)$. The mapped additivity can be written as*

$$g(T(\mathcal{X}|Q)) = \sum_{j=1}^{M} g(T(\mathcal{X}_{m_j}^{n_j}|Q_{r_j}^{u_j})), \quad \textit{for any } Q, 0 < M \leq L, \mathcal{X}. \tag{6}$$

Lemma 3.3. *(Optimal sub-quantizer) Let quantizer Q^*, among all the quantizers that have L quantization levels, maximize the performance measure $T(\mathcal{X}|Q)$ on the data set $\mathcal{X}$ of size N. Let x_n be the largest element in interval q of quantizer Q^*. Then the sub-quantizer Q_1^{*q}, among all the sub-quantizers that have q quantization levels and x_n as their largest element in interval q, maximizes the performance measure $T(\mathcal{X}_1^n|Q_1^q)$, i.e. $T(\mathcal{X}_1^n|Q_1^{*q}) = \max_{Q_1^q} T(\mathcal{X}_1^n|Q_1^q)$.*

Proof by contradiction. By the mapped additive property of T,

$$g(T(\mathcal{X}|Q^*)) = g(T(\mathcal{X}_1^n|Q_1^{*q})) + g(T(\mathcal{X}_{n+1}^N|Q_{q+1}^{*L}))$$

Since x_n is always the largest element of interval q, the second term $T(\mathcal{X}_{n+1}^N|Q_{q+1}^{*L})$, which is the performance measure in the last $L-q$ intervals on data $\{x_{n+1}, \cdots, x_N\}$, would not be affected by the choice of Q_1^{*q}.

Assume that $\hat{Q}_1^q$ was another sub-quantizer that quantizes $\mathcal{X}_1^n$ into q intervals with x_n being the largest element in interval q that does better in performance than Q_1^{*q}, that is,

$$T(\mathcal{X}_1^n|\hat{Q}_1^q) > T(\mathcal{X}_1^n|Q_1^{*q}). \tag{7}$$

We could create a new quantizer $\hat{Q}$ by combining the sub-quantizer $\hat{Q}_1^q$ and Q_{q+1}^{*L}, which has the performance measure

$$\begin{aligned} & g(T(\mathcal{X}|\hat{Q})) \\ =& g(T(\mathcal{X}_1^n|\hat{Q}_1^q)) + g(T(\mathcal{X}_{n+1}^N|Q_{q+1}^{*L})) \\ >& g(T(\mathcal{X}_1^n|Q_1^{*q})) + g(T(\mathcal{X}_{n+1}^N|Q_{q+1}^{*L})) \\ =& g(T(\mathcal{X}|Q^*)). \end{aligned}$$

By the monotonically increasing property of $g(x)$, the above leads to

$$T(\mathcal{X}|\hat{Q}) > T(\mathcal{X}|Q^*).$$

This conclusion contradicts the condition that $T(\mathcal{X}|Q^*)$ is the maximum performance measure on $\mathcal{X}_1^N$ among all quantizers with L levels. Then the assumption made in Eq. (7) must be incorrect. Thus

$$T(\mathcal{X}_1^n|Q_1^{*q}) \geq T(\mathcal{X}_1^n|\hat{Q}_1^q) \tag{8}$$

must be true. Therefore, $T(\mathcal{X}_1^n|Q_1^{*q})$ maximizes the performance measure on the subsequence $\mathcal{X}_1^n$ over q quantization levels, that is,

$$T(\mathcal{X}_1^n|Q_1^{*q}) = \max_{Q_1^q} T(\mathcal{X}_1^n|Q_1^q).$$

□

Next, we establish the optimality of quantization by dynamic programming under the mapped additivity condition.

Theorem 3.4. *If $T(\mathcal{X}|Q)$ satisfies the mapped additivity condition defined in Eq. (6), finding an optimal quantization Q^* of L levels to maximize $T(\mathcal{X}|Q)$ can be solved exactly using dynamic programming by the recurrence*

$$(9)\ T[n,q] = \begin{cases} 0 & n = 0 \text{ or } q = 0 \\ \max\limits_{1 \le i \le n} T[i-1, q-1] + g(T(\mathcal{X}_i^n | Q_q^q)), & 1 \le n \le N,\ 1 \le q \le L, \end{cases},$$

and the optimal performance measure is

$$T(\mathcal{X}|Q^*) = \max_{Q} T(\mathcal{X}|Q) = g^{-1}(T[N,L]).$$

Proof. By the recursive definition of $T[n,q]$ in Eq. (9), we must have

$$T[n,q] = \max_{Q_1^q} g(T(\mathcal{X}_1^n | Q_1^q)),$$

due to Lemma 3.3, i.e., $T[n,q]$ must correspond to the optimal mapped performance measure that can be achieved for the first n points over q quantization levels. Thus $T[N,L]$ corresponds to the optimal performance measure for the entire data set with L quantization levels. Therefore, the inversely mapped value $g^{-1}(T[N,L])$ achieves the optimal performance measure $T(\mathcal{X}|Q^*)$ obtained by an optimal quantizer Q^*. □

With $g(x) = x$ and under the constraint that a decision boundary in Q must be a middle point between some pair of consecutive distinct points, $T(X|Q)$ as shown in Eq. (5) meets the mapped additivity requirement. In addition to our definition of $T(X|Q)$, many problems in data mining involve performance measures that satisfy such a condition. Examples include k-means clustering operating in any metric space, and discretization that maximizes classification accuracy using either class purity entropy or percentage of correct classifications.

4 Maximum Likelihood Quantization using Dynamic Programming

As the optimality condition Eq. (6) holds for $T(X|Q)$, we can use dynamic programming to find an optimal quantization that maximizes $T(X|Q)$. To avoid over-fitting, we require a minimum number of k data points in each bin and that identical ones are put into the same bin. We only set a decision boundary in the middle of two consecutive and distinct data points. This affects the range of $J(X|Q)$, but it is trivial when the sample size is not too small. This restriction prevents $J(X|Q)$ from overflow. Let T be an $(N+1)\times(L+1)$ matrix, whose entry $T[n,q]$ $(0 \le n \le N, 0 \le q \le L)$ is the maximum performance measure from bin 1 to q when x_n is the largest data in bin q. Let I be an $(N+1)\times(L+1)$ matrix, whose entry $I[n,q]$ $(0 \le n \le N, 0 \le q \le L)$ is the index to the smallest element in bin q such that $T[n,q]$ is achieved. Let T^1 be an $N \times N$ matrix, whose entry $T^1[i,n]$ $(1 \le i \le n \le N)$ is the performance measure contributed by a sub-quantizer with a single bin containing exactly x_i to x_n, that is,

$$T^1[i,n] = T(\mathcal{X}_i^n | Q_q^q), \quad \forall q \in \{1,2,\ldots,L\}$$

The dynamic programming for finding a quantization to maximize $T[N,L]$ is described below.

Initialization – $T[n,q]$ is set to zero when either no point is covered ($n = 0$) or no quantization is applied ($q = 0$) as in Eq. (10). $I[n,q]$ is initialized as in Eq. (11): $I[0,0] = 0$ indicates the halting of backtrack; The -1 values indicate that those locations are invalid as the quantization would either on an empty

set, or there would be more levels than points, or some bins would be empty.

$$T[n,q] = 0, \quad n = 0 \text{ or } q = 0 \tag{10}$$

$$I[n,q] = \begin{cases} 0, & n = 0, q = 0 \\ -1, & n = 0, q > 0; \text{ or } n > 0, q = 0 \\ -1, & 0 \le q < \max(1, n - (N - L)), n \ne 0, q \ne 0 \\ -1, & \min(n, L) < q \le L, n \ne 0, q \ne 0 \end{cases} \tag{11}$$

Feasible decision boundary index set – The indices of the feasible data for being the smallest element in bin q form the feasible decision boundary index set

$$\mathcal{A}_q^n = \{i | i \le n-k+1, I[i-1, q-1] \ne -1, x_{i-1} \ne x_n, I[n,q] \ne -1, x_n \ne x_{n+1}\}.$$

The inequality $i \le n - k + 1$ guarantees that at least k data points are in bin q; $I[i-1, q-1] \ne -1$ states that x_{i-1} must be feasible for the largest point in the previous bin $q-1$; $x_{i-1} \ne x_n$ enforces that the feasible largest point in the previous bin $q-1$ must not be the same as x_n, to avoid splitting identical data points into different bins; $x_n \ne x_{n+1}$ is also not to split identical data points; $I[n,q] \ne -1$ asserts that x_n must be feasible for the largest point of bin q.

Recurrence – If $\mathcal{A}_q^n$ is empty, then $I[n,q] \triangleq -1$, meaning x_n does not qualify for the largest point in bin q. Otherwise,

$$T[n,q] \triangleq \max_{i \in \mathcal{A}_q^n} T[i-1, q-1] + T^1[i,n], \tag{12}$$

$$I[n,q] \triangleq \operatorname*{argmax}_{i \in \mathcal{A}_q^n} T[i-1, q-1] + T^1[i,n]. \tag{13}$$

Algorithm 1 Find-Optimal-Quantization fills matrices T and I row by row using the recurrence equations. The range limit of q in line 5 is equivalent to filling the

lower left and upper left corners of matrix I with -1. The actual initialization of the first column of I is implicit from line 7 to 12. Line 15 decides the feasible decision boundary set. Lines 17 and 18 implement the recurrence equation if $\mathcal{A}$ is not empty. Matrix I is returned for backtracking.

Algorithm 1 Find-Optimal-Quantization($\mathcal{X}$, L, k)

```
1: Sort X in non-decreasing order if not already so
2: Initialize T and I;
3: for n ← 1 to N do
4:    Calculate the n-th column of T^1;
5:    for q ← max(1, n − (N − L)) to min(n, L) do
6:       if n ≠ N and x_n = x_{n+1} then
7:          I[n, q] ← −1;
8:       else if q = 1 then
9:          if n ≥ k then
10:            T[n, q] ← T^1[1, n], I[n, q] ← 1;
11:         else
12:            I[n, q] ← −1;
13:         end if
14:      else
15:         A ← {i | q ≤ i ≤ n − k + 1, x_{i−1} ≠ x_n, I[i − 1, q − 1] ≠ −1, x_n ≠ x_{n+1}};
16:         if A ≠ ∅ then
17:            T[n, q] ← max_{i∈A} T[i − 1, q − 1] + T^1[i, n];
18:            I[n, q] ← argmax_{i∈A} T[i − 1, q − 1] + T^1[i, n];
19:         else
20:            I[n, q] ← −1;
21:         end if
22:      end if
23:   end for
24: end for
25: return I;
```

Once matrix I is determined, an optimal quantization can be retrieved by Alg. 2 Backtrack($\mathcal{X}$, I). Backtracking starts from $I[N, L]$ and traces back to $I[0, 0]$. Two dummy data points $-\infty$ and $+\infty$ are introduced in line 2. If a finite range quantizer is needed, we can set them to $x_1 - \delta$ and $x_N + \delta$ instead, where δ is a quantity not larger than the data resolution. When the performance measure contains the average log likelihood, we shall use finite width intervals. Since the value of $I[n, q]$ is the index to the smallest point in interval q if x_n is the largest point of interval q, $I[n, q] - 1$ must be the index to the largest point in interval $q - 1$. The backtrack

proceeds until $q = 0$. Each decision boundary is set to the middle of two adjacent points in different intervals (line 4).

Algorithm 2 Backtrack($\mathcal{X}$, I)

1: $n \leftarrow N$, $q \leftarrow L$;
2: $x_0 \leftarrow -\infty$, $x_{N+1} \leftarrow +\infty$;
3: **while** $q \neq 0$ **do**
4: $\quad b_q \leftarrow \frac{x_n + x_{n+1}}{2}$;
5: $\quad n \leftarrow I[n, q] - 1$, $q \leftarrow q - 1$;
6: **end while**
7: $b_0 \leftarrow x_0$;
8: **return** Q;

Theorem 4.1. *The dynamic programming algorithm (Alg. 1) has time complexity $O(LN^2)$. The backtrack algorithm (Alg. 2) has time complexity $O(L)$.*

Proof. $O(N \log N)$ is used in sorting the data. $O(LN^2)$ is used for filling in matrix T and I. Brute force calculation of matrix T^1 can take up to $O(N^3)$, immediately making the algorithm impractical to use when N is moderately large. Since $T^1[i, n]$ can be calculated from its neighbor $T^1[i-1, n]$ or $T^1[i+1, n]$ in constant time with minor memory costs, only $O(N^2)$ is used for filling in matrix T^1. So Alg. 1 Find-Optimal-Quantization has $O(N \log N + N^2 + LN^2) = O(LN^2)$ as its overall time complexity.

$O(L)$ is spent backtracking the optimal intervals, since the while-loop has exactly L iterations and within each iteration it takes constant time. □

Theorem 4.2. *The dynamic programming algorithm (Alg. 1) has space complexity $O(LN)$.*

Proof. In the most straightforward implementation, $N(N+1)/2$ would be needed to store matrix T^1, which can actually be reduced to linear space. When the n-th rows of T and I are calculated, only the n-th column of T^1 is used and this column will not be used again. Thus, during any iteration of the for-loop on n, we save only the n-th column of T^1. This will reduce the space needed for T^1 from N^2 to

N. We need $2LN$ space for matrices T and I. So the total space complexity is $O(2LN + N) = O(LN)$, which is the original claim. □

The dynamic programming algorithm taking sample points can be readily changed to apply to merge counts of data already accumulated in finer-than-desired bins, because the performance measure uses only counts of data within a bin and the bin widths rather than the actual values of those points.

5 Estimation of Recombination Rate Distribution over Chromosomes by Quantization

Recombination is a biological phenomenon that is of central importance to the fields of genetics and evolutionary biology. In the nucleus of each human cell (except the haploid gametes) each chromosome (except the sex chromosomes) comes in two copies called homologous chromosomes, one chromosome coming from the mother and one from the father. During meiosis (that is the formation of four haploid gametes from a diploid cell) homologous chromosomes exchange their genetic materials in a process called recombination. Thus, the chromosomes at the next generation do not contain the same genetic information as the parent's chromosomes but instead are a mosaic of alleles from the mother's and father's chromosomes. The study of recombination is important to the field of molecular evolution because the local rate of recombination affects the efficiency of natural selection. *Recombination rate* (RR) is defined as the number of recombination events in a unit length of chromosome in terms of base pairs (bps), usually in centiMorgan per Mbps (cM/Mb). The RR distribution (RRD) function maps a location on the chromosome to an RR value. However, observing recombination events has been limited due to the cost of experiments. As the complete human genome physical map becomes available, an accurate quantitative representation of the RRD becomes possible.

Recombination events are identified using both genetic and physical maps. On a genetic map, each marker represents a unique feature. A marker has two or

multiple forms, called *alleles*. The alleles can be identified by polymerase chain reactions. Locations of markers on the physical map are determined in advance. Markers make detection of recombination events possible without sequencing the entire genomes of generations. The resolution of the identified events increases with the number of markers. This method is illustrated in Fig. 2. The first parent has 2 markers A and B (Fig. 2(a)) and the second parent has the same markers but with different alleles a and b (Fig. 2(b)). If a child has the markers as in Fig. 2(c), then at least one recombination event has occurred at some location between markers A and B. If a child has the same alleles as their parents as in Fig. 2(d), then it is unlikely to have a recombination event between A and B if the markers are close enough. This method cannot detect the exact location of a recombination

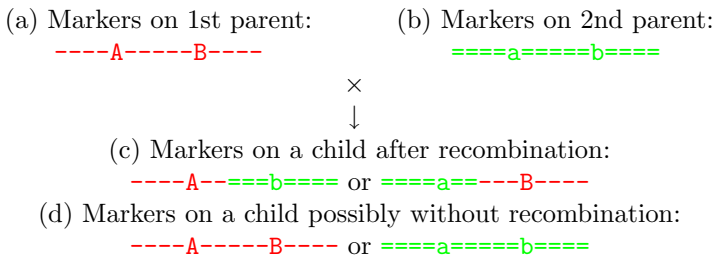


Figure 2: Identifying a recombination event with markers. One marker has two alleles A and a; the other has two alleles B and b.

event or it may miss a recombination event between markers. In addition, if the two parents carry the same set of alleles, no recombination event between the markers may be identified. Therefore, selection of markers directly affects the effectiveness of recombination detection. Typically, a good marker collection should be abundant, hyper-variable, and evenly distributed across the genome. One such family of markers is microsatellites, which are short sequences of motifs in tandem (Brown, 1999). The motifs can be di-, tri-, or tetra-nucleotide repeat units. In the Marshfield recombination map (Yu and et al., 2001), over 8,000 microsatellites are used; in the Iceland recombination map (Kong and et al., 2002), there are 5,000 microsatellites.

The frequency of recombination is not uniform across the genome: More frequent near the *telomere* – the end of a eukaryotic chromosome – and less frequent at the *centromere* where two copies of the homologous chromosomes hold together. We consider X, the location of a recombination event, a random variable. Let $p(x)$ be its p.d.f. Let $F(x)$ be its cumulative distribution function (c.d.f.).

The RRD function $R(x)$ is in proportion to $p(x)$ defined as $R(x) = R_0 p(x)$, where R_0 is the total amount of recombination events observed on a single chromosome of an individual. This definition is used in the Iceland RRD estimation (Kong and et al., 2002). Since its exact physical location is unknown, a recombination event between two markers is assigned the position of the marker with larger coordinate on the chromosome. With N recombination event locations x_1, x_2, $\cdots$, x_N observed, an estimated p.d.f. $\hat{p}(x)$ is obtained using the Parzen window method in (Kong and et al., 2002)

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x, x_i), \tag{14}$$

where

$$k(x, x_i) = \begin{cases} \frac{1}{\Delta}, |x - x_i| \leq \frac{\Delta}{2} \\ 0, \quad \text{otherwise} \end{cases},$$

and Δ is the bandwidth. Then they choose a sequence of M equally spaced locations $y_0, 2y_0, 3y_0, \cdots, My_0$ to calculate the estimated p.d.f. values. In the end, they fit splines to these points to obtain a smooth p.d.f $p(x)$ and then obtain $R(x)$. The critical bandwidth parameter Δ is 3 Mbps. The sample is drawn from 1257 meioses.

Another RRD is defined by $R(x) = R_0 \frac{dF(x)}{dx}$, used by the Marshfield RRD (Yu and et al., 2001). In this approach, it is not necessary to know the exact location of each recombination event. They compute the empirical c.d.f. $\hat{F}(x)$ from the observed recombination events, then fit cubic splines to $\hat{F}(x)$ and then obtain the RRD. In this study, only 184 meioses are analyzed to identify recombination events, which is a much smaller sample size compared to (Kong and et al., 2002).

The RRDs in (Kong and et al., 2002) are represented as continuous functions, with empirically chosen bandwidth Δ. All the splines are saved and must be evaluated to calculate RRD at a location.

Alternatively, we performed optimal quantization on the genetic distances of selected markers (Kong and et al., 2002), given as the empirical c.d.f. of the recombination events. We first obtained the control parameters w_J, w_H, L, and k by a 5-fold cross-validation. The values of w_J and w_H range from 0 to 1 with a step of 0.1. L ranges from 2 to 2^8 in powers of 2. k ranges from 1 to 3^6 in powers of 3. Second, using the best parameters, a p.d.f. was estimated, on all the recombination events for each chromosome. The estimated RRD functions of chromosomes 3 and X are shown in Fig. 3 and 4. Recombination is much more active around the ends of chromosomes than the centers. Our RRDs show more fluctuations than those shown in (Kong and et al., 2002; Yu and et al., 2001). Since our control parameters are all cross-validated, it is very likely that the RRDs indeed change more abruptly than the much more smooth curves published before. To fit splines on our estimation result could make the curve smoother, but it requires validation of the smoothness. We further compare quantitatively the performance of optimal quantization with the Parzen window method. To make the comparison fair, we did not apply splines. The evaluation is done by a 5-fold cross-validation. The performance measure is the log likelihood of the left-out data reserved for test, using the p.d.f. estimated from the data not using the left-out data. The average and the standard deviation of the cross-validated log likelihood for each chromosome are shown in Table 1. The average log likelihoods of the p.d.f. obtained by optimal quantization are consistently higher than those by the Parzen window method. The standard deviations of both are similar, with Parzen window results slightly smaller on most of the chromosomes. Therefore the optimization quantization approach provides a better RRD estimation than that of the Parzen window.

Table 1 Comparison between optimal quantization & Parzen window.

Chromosome	Average Log Likelihood		Standard Deviation	
	Quantization	Parzen Window	Quantization	Parzen Window
1	-19.11	-19.17	0.03	0.01
2	-19.10	-19.21	0.05	0.02
3	-18.90	-19.05	0.04	0.04
4	-18.91	-18.98	0.03	0.02
5	-18.79	-18.91	0.04	0.03
6	-18.72	-18.88	0.05	0.03
7	-18.69	-18.87	0.03	0.02
8	-18.60	-18.78	0.02	0.01
9	-18.42	-18.52	0.04	0.03
10	-18.55	-18.69	0.05	0.05
11	-18.53	-18.65	0.06	0.03
12	-18.57	-18.63	0.03	0.04
13	-18.02	-18.32	0.06	0.04
14	-17.94	-18.14	0.07	0.07
15	-17.87	-18.17	0.06	0.07
16	-18.05	-18.18	0.07	0.04
17	-17.99	-18.14	0.05	0.05
18	-18.04	-18.16	0.08	0.06
19	-17.70	-17.95	0.09	0.05
20	-17.62	-17.70	0.09	0.03
21	-17.05	-17.28	0.06	0.05
22	-16.96	-17.16	0.08	0.05
X	-18.42	-18.53	0.04	0.03

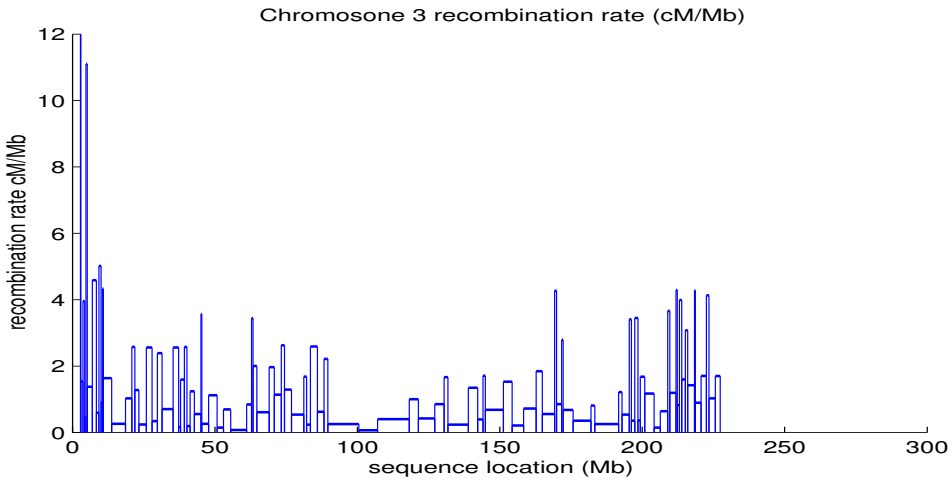


Figure 3: Chromosome 3.

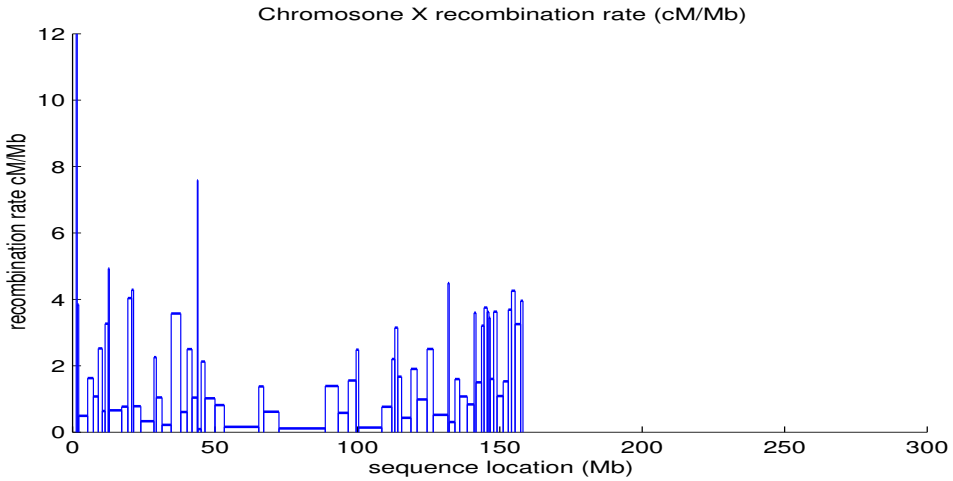


Figure 4: Chromosome X.

6 Localized Study of Recombination Rate within Length Groups of L1s

L1 retrotransposons have significantly affected the structure and function of mammalian genomes, including the human genomes. They have been a source of genetic novelty and their activity accounts for at least 30% of the size of our genome. However, their replicative success is difficult to reconcile with the potential damages they can impose on their host's genome. The effect that L1 elements can have on the fitness of individuals remains a matter of debate. One approach used to understand their impact is to look at their distribution in the genome relative to the local recombination, The rationale is that if L1 elements of a given length are deleterious they should accumulate in regions of low recombination.

Therefore we decided to examine how RR near an L1 element depends upon the length of the element. A linear regression could not adequately capture subtlety of the RR-length interaction. Given the relatively large sample size of L1s, instead of fitting a higher order linear regression model, we analyze families of different age separately using the classification of Khan et al. (2006). We studied five families, named L1PA2 to L1PA6, and broke elements within each family into groups based on the length of the elements. We then looked at the trend of RR within each

group. Grouping is determined by optimal quantization of the lengths of all L1s under consideration. Intuitively, this method separates L1s into groups by length when there is a sudden change in the number of L1s over unit length. We selected the number of groups to be six, roughly capturing the overall distribution of length while assuring that the intervals are not too small for a meaningful regression. The six length groups are shown in Table 2. The grouping reflects a natural tendency for L1 to segregate by length.

Table 2 L1 groups by length, with length ranges, counts, and percentage.

L1 Groups	Length Range	L1 Count/Percentage
1	[100,490]	12226/34%
2	[491,1152]	8559/24%
3	[1153,2498]	6462/18%
4	[2499,6001]	4182/12%
5	[6002,6183]	4231/12%
6	≥ 6184	218/1%

A one-way ANOVA (Table 3) indicates indeed the RR means are significantly different among L1 length groups. The Tukey's Honest Significant Differences

Table 3 One-way ANOVA for RR over the length groups.

	Degrees of Freedom	Sum of Squares	Mean Squares	F value	$\Pr(>F)$
group	5	93	19	7.5441	4.330e-07
Residuals	35872	88107	2		

(HSD) test reveals further details in Fig. 5. Under the null hypothesis of RR mean equality across groups, if one compares every two groups using the 5% α-level, the chance of observing some inequality among the pairs can be much greater than the anticipated 5% type I error. The Tukey's HSD test corrects this problem. In Fig. 5, the range of each line segment manifests the 95% confidence interval of the mean RR difference between the two length groups labeled on the left of the segment. The vertical dashed line marks the zero difference location. If an interval contains zero, there is no significant evidence from the sample to conclude that the two groups have different mean RRs. All differences are the mean RR of a group with a longer length minus that of one with a shorter length. A major observation is that no segments have both ends above zero, suggesting no significant trend of

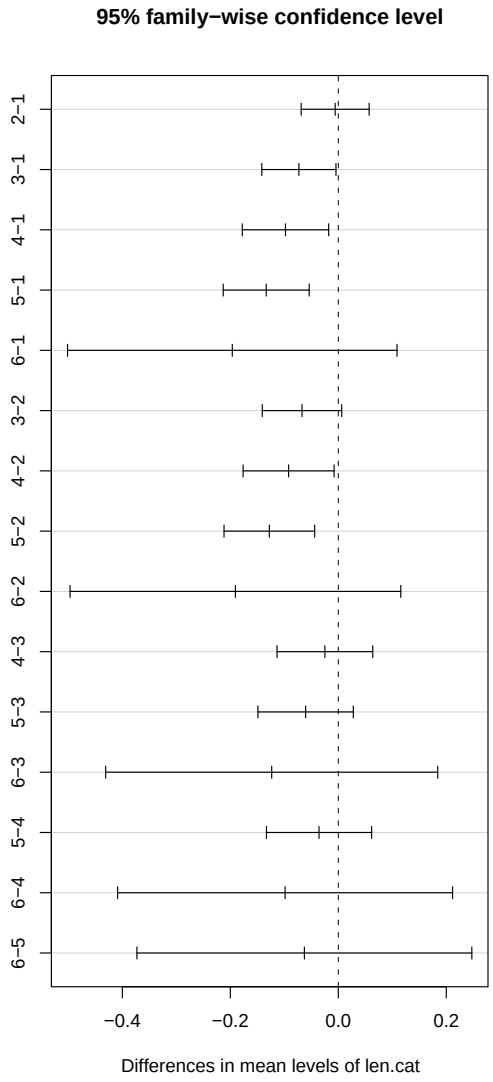


Figure 5: Tukey's HSD test on the RR means among length groups. Numbers on the vertical axes correspond to length groups. For example, 5-3 stands for the mean RR of group 5 minus that of group 3.

increasing RR as length increases. The only almost significant negative difference between two consecutive length groups occurs from group 2 to 3, which accounts for other significant differences among non-consecutive length groups. Therefore, the multiple comparison analysis pins down that the most significant reduction in

RR takes place among the L1s of intermediate length, that is between elements shorter and longer than 1.2Kb.

Based on the Tukey's HSD results, we studied the trend of RR within each length group using linear regression on the length of L1. The intercepts and slopes of each linear regression line, and the corresponding p-values are given in Table 4. No length group shows a significant positive slope. We observe that length group 2 has a highly significant negative slope. Figure 6 shows the mean RR-length scatter plot with the regression lines overlaid. We can observe in the plot a decreasing trend of the regression line in group 2 quite evidently. It is also quite evident subjectively that there is a declining tendency in the mean RR as the length increases. This further analysis match well to previous findings by the Tukey's HSD test. Therefore the major RR reduction occurs on the L1s of length 491 to 1152, which are not full-length L1s, but L1s of intermediate length.

Table 4 Linear regression slopes of each group.

	Estimate	Std. Error	t-Statistic	$\Pr(>\|t\|)$
1:length	-5.537e-05	1.275e-04	-0.434	0.6641
2:length	-2.446e-04	9.006e-05	-2.716	0.0066
3:length	-3.409e-05	5.126e-05	-0.665	0.5060
4:length	3.042e-06	2.268e-05	0.134	0.8933
5:length	5.923e-05	4.409e-04	0.134	0.8931
6:length	3.108e-04	5.386e-04	0.577	0.5639

7 Conclusion

We have described a dynamic programming algorithm to quantize a random variable to preserve maximally the p.d.f. of the original continuous variable. Although our algorithm has a quadratic running time in sample size, it guarantees the optimality of quantization. The distance-based k-means algorithm for univariate quantization, popular simply due to its computational convenience, shall either be replaced by our maximum likelihood approach when preservation of the distribution of the original continuous random variable is desired, or by a dynamic programming implementation similar to ours that guarantees optimality. Applications of

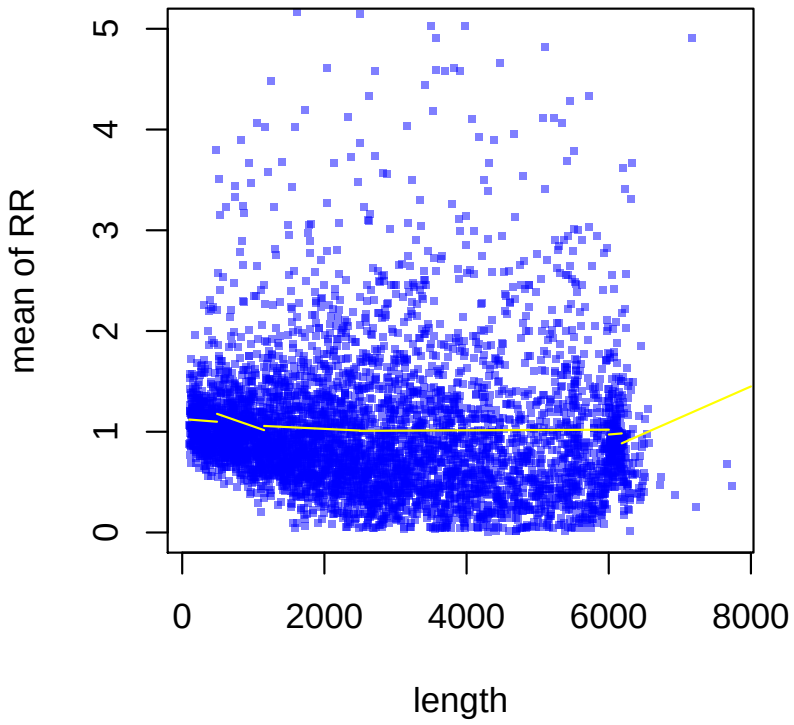


Figure 6: Scatter plot of mean RR versus L1 length. The line segments are linear regressions within each group. Only the second segment has a significant decreasing trend.

our algorithm in estimating RR distributions and characterizing L1 elements show its effectiveness in capturing the underlying p.d.f.s of data. It can also be used to discretize other genomic features including GC-content, gene expression rate, and non-coding element densities over a genome.

Acknowledgements

The authors thank the support from grants made by PSC-CUNY, CUNY Institute for Software Design and Development, and NSF CREST Center for Excellence in Computational Biology and Bioinformatics (Grant Number: HRD_0420407).

Proof: TBD. To be determined

References

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth & Brooks/Cole, Pacific Grove, California.

Brown, T. A. (1999). *Genomes*, Wiley-Liss.

Fulton, T., Kasif, S. and Salzberg, S. L. (1995). Efficient algorithms for finding multi-way splits for decision trees, *Proc. 12th Int'l Conf. on Machine Learning*, pp. 244–251.

Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*, Kluwer Academic Publishers.

Haralick, R. M. (1976). The table look-up rule, *Communications in Statistics – Theory and Methods* **A5**(12): 1163–91.

Haralick, R. M., Shanmugam, K. and Dinstein, I. (1973). Textural features for image classification, **SMC-3**(6): 610–621. See Appendix for equal-probability quantization.

Hearne, L. B. and Wegman, E. J. (1992). Maximum entropy density estimation using random tessellations, *Computing Science and Statistics*, Vol. 24, pp. 483–7.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliff, NJ.

Khan, H., Smit, A. and Boissinot, S. (2006). Molecular evolution and tempo of amplification of human LINE-1 retrotransposons since the origin of primates, *Genome Research* **16**: 78–87.

Kong, A. and et al. (2002). A high-resolution recombination map of the human genome, *Nature Genetics* **31**: 241–247.

Margaritis, D. and Thrun, S. (2001). A Bayesian multiresolution independence test for continuous variables, *17th Conference on Uncertainty in Artificial Intelligence (UAI)*, Seattle, Washington.

Scott, D. W. (1992). *Multivariate Density Estimation – Theory, Practice and Visualization*, John Wiley & Sons.

Voronoi, G. (1908). Nouvelles applications des parametres continus a la théorie des formes quadratiques, deuxieme memoire, recherches sur les parallelloedres primitifs, *Journal für die Reine und Angewandte Mathematik* **134**(198-287).

Wu, X. (1992). Color quantization by dynamic programming and principal analysis, *ACM Trans. Graph.* **11**(4): 348–372.

Yu, A. and et al. (2001). Comparison of human genetic and sequence-based physical maps, *Nature* **409**: 951–953.