Estimating Change-Points in Biological Sequences via the Cross-Entropy Method

G. E. Evans, G. Yu. Sofronov, J. M. Keith and D. P. Kroese

September 20, 2010

Abstract

The genomes of complex organisms, including the human genome, are known to vary in GC content along their length. That is, they vary in the local proportion of the nucleotides G and C, as opposed to the nucleotides A and T. Changes in GC content are often abrupt, producing well-defined regions.

We model DNA sequences as a multiple change-point process in which the sequence is separated into segments by an unknown number of change-points, with each segment supposed to have been generated by a different process. Multiple change-point problems are important in many biological applications, particularly in the analysis of DNA sequences. Multiple change-point problems also arise in segmentation of protein sequences according to hydrophobicity.

We use the Cross-Entropy method to estimate the positions of the change-points. Parameters of the process for each segment are approximated with maximum likelihood estimates. Numerical experiments illustrate the effectiveness of the approach. We obtain estimates of the locations of change-points in artificially generated sequences and compare the accuracy of these estimates with those obtained via other methods such as IsoFinder [1] and Markov Chain Monte Carlo. Lastly, we provide examples with real data sets to illustrate the usefulness of our method.

1 Introduction

This paper considers the problem of identifying change-points in a very long binary sequence. In this context, a *change-point* is a position in the sequence such that the frequency of the '1' character differs on either side. To explain why this is an important problem and why it is of interest, it is necessary to first introduce a small amount of biology.

The genomes of complex organisms (e.g., humans), contain vast amounts of information specifying components of cellular systems and control mechanisms for regulating their interactions. This information is encoded in long linear molecules of DNA, which are comprised of four nucleotides, or 'characters', denoted A, C, G and T. One important class of components in cellular systems is that of proteins. Proteins are composed of amino acids, of which there are 20 main types. Molecular machines in the cell are able to translate the 4-character alphabet of nucleotides into the 20-character alphabet of amino acids, thus manufacturing proteins. However, only about 1% of the human genome encodes for proteins. While much of the remainder may be 'junk DNA' (performing no function of any importance to the organism), it is widely believed that at least 5% of the human genome is functional [3]. While the protein-coding portion of the human genome is almost completely identified, the functional non-protein-coding portion is poorly understood and is only beginning to be characterised. It is likely that entire classes of functional RNAs remain to be discovered [7]. Major biological advances are likely to result from the study of this component.

There is a scarcity of effective methods for discovering functional RNAs in genomes. The main reason for this is simply that so little is known about them that it is unclear what to look for. We are pursuing a non-hypothesis based approach, in which we identify key sequence characteristics that may be indicative of function, and then look for positions in the genome where these properties change discontinuously. These change-points are then considered as putative boundaries of functional non-protein-coding RNAs. Other experimental and analytic approaches can then be employed to verify the presence of a functional element.

Various properties of genomic sequence that are potentially indicative of function can be represented as a binary sequence. One property that is useful for considering the problem in the abstract (though by no means the best indicator of function) is GC content. Positions in a DNA sequence at which a G or C nucleotide is present are represented by a 1, whereas positions at which an A or T is present are represented by a 0. The reason for considering G and C together is that DNA is actually a double-stranded molecule in which a G on one strand pairs opposite a C on the other strand, and an A on one strand pairs opposite a T on the other.

The approach taken here uses Bayesian sequence segmentation to identify putative change-points in a binary sequence. This technique has been extensively studied. See [2] for a review of older methods, and papers by Keith et al. [5, 6, 10] for more recent developments.

In this paper we present a *Cross-Entropy* [8] approach to change-point modeling, using Monte Carlo simulation to find estimates of change-points. We include results of numerical experiments indicating the usefulness of this method. We apply the method to real data from the human genome to detect segmental variation in GC content, but the method could equally be applied to detect segmental variation in other important situations.

The paper is structured as follows: Section 2 includes a statement of the multiple change-point problem in mathematical terms. In Section 3, we explain the basic framework of the Cross-Entropy method. In Section 4, we develop the Cross-Entropy method for the multiple change-point problem. Section 5 presents the results of two numerical experiments.

2 The Multiple Change-Point Problem

Let us formulate the multiple change-point problem (MCPP) in mathematical terms. A binary sequence $\mathbf{b} = \{b_1, \dots, b_L\}$ of length L is given. A segmentation of the sequence is specified by giving the number of change-points N and the positions of the change-points $\mathbf{c} = (c_1, \dots, c_N)$, where $0 = c_0 < c_1 < \cdots < c_N < c_{N+1} = L$. In this context, a change-point is a boundary between two adjacent segments, and the value c_i is the sequence position of the rightmost character of the segment to the left of the i-th change-point. Segments are numbered from 0 to N as there will be one more segment than change-points. A maximum number of change-points d is specified, where $0 \leq N \leq d < L$. The model assumes that within each segment, characters are generated via independent Bernoulli trials with probability of success (that is obtaining a "1") θ_n, where $0 < \theta_n < 1$ for $n = 0, \dots, N$. Then the joint distribution of $b_1, \dots, b_L$ conditional on N, $\mathbf{c} = (c_1, \dots, c_N)$, and $\boldsymbol{\theta} = (\theta_0, \dots, \theta_N)$ is given by

$$\begin{aligned} & f(b_1, \dots, b_L \mid N, \mathbf{c}, \boldsymbol{\theta}) \\ & = \prod_{n=0}^{N} \theta_n^{\mathbb{I}(c_n, c_{n+1})} (1 - \theta_n)^{\mathbb{O}(c_n, c_{n+1})}, \end{aligned}$$

where

$$\begin{aligned} \mathbb{I}(c_n, c_{n+1}) &= \sum_{i=c_n+1}^{c_{n+1}} b_i, \\ \mathbb{O}(c_n, c_{n+1}) &= c_{n+1} - c_n - \mathbb{I}(c_n, c_{n+1}). \end{aligned}$$

In other words, $\mathbb{I}(c_n, c_{n+1})$ is the number of ones in the segment bounded by sequence positions $c_n + 1$ and c_{n+1} and $\mathbb{O}(c_n, c_{n+1})$ the number of zeros in that same segment.

To formulate the problem in terms of a Bayesian model, we use the framework set out in [4]. Let $\mathcal{X}$ be the set of possible values of $\mathbf{x} = (N, \mathbf{c}, \boldsymbol{\theta})$, where $\mathcal{X} = \cup_{N=0}^{d} \left[\{N\} \times \mathcal{C}_N \times (0,1)^{N+1} \right]$, with $\mathcal{C}_N = \{(c_1, \dots, c_N) \in \{1, \dots, L - 1\}^N : c_1 < \dots < c_N\}$. We assume a uniform prior both on the number of change-points and on $\mathcal{C}_N$, and uniform priors on $(0,1)$ for each θ_n. Thus, the overall prior $f_0(N, \mathbf{c}, \boldsymbol{\theta})$ is constant. The use of uniform priors means that this Bayesian setting is equivalent to the Maximum Likelihood approach. However, if one had a more informative priors, they could be used instead of the uniform priors. The posterior distribution at point $\mathbf{x} = (N, \mathbf{c}, \boldsymbol{\theta})$, having observed $b_1, \dots, b_L$, is given by

$$\begin{aligned} \pi(\mathbf{x}) \quad & \propto f_0(N, \mathbf{c}, \boldsymbol{\theta}) f(\mathbf{b} \mid N, \mathbf{c}, \boldsymbol{\theta}) \\ & = \prod_{n=0}^{N} \theta_n^{\mathbb{I}(c_n, c_{n+1})} (1 - \theta_n)^{\mathbb{O}(c_n, c_{n+1})}. \end{aligned}$$

3 The Cross-Entropy Method

The *Cross-Entropy* (CE) method [8] can be used for two types of problems:

- Estimation,
- Optimization.

Suppose we wish to solve the following maximization problem: Let $\mathcal{X}$ be a finite set of *states* and S a real-valued *performance function* on $\mathcal{X}$. We wish to find the maximum value of S over $\mathcal{X}$ and the state(s) corresponding to this value. Let γ^* be the maximum of S over $\mathcal{X}$ and let $\mathbf{x}^*$ be a state at which this maximum is attained. Then,

$$S(\mathbf{x}^*) = \gamma^* = \max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x}). \tag{1}$$

The CE method is an iterative optimization method that starts with a parameterized sampling distribution $f(\mathbf{x}; \mathbf{u})$ from which a random sample is generated. Each observation in this sample is *scored* for its performance as the solution to a specified optimization problem. A fixed number of the best of these observations are referred to as the *elite sample.* This elite sample is used to update the parameters for the sampling distribution. The sampling distribution eventually converges to a degenerate distribution about a locally optimal solution which ideally will be globally optimal.

The first step of the CE method is to turn the optimization problem (1) into a meaningful *estimation problem.* Let $I_{\{S(\mathbf{X}) \geq \gamma\}}$ be a collection of indicator functions for various levels γ. Then, (for the discrete case) we associate the estimation of

$$\ell(\gamma) = \mathbb{P}_{\mathbf{u}}(S(\mathbf{X}) \geq \gamma) = \sum_{\mathbf{x}} I_{\{S(\mathbf{x}) \geq \gamma\}} f(\mathbf{x}; \mathbf{u}) = \mathbb{E}_{\mathbf{u}} \left[I_{\{S(\mathbf{X}) \geq \gamma\}} \right]$$

with (1). Now we use a two-part iterative approach to obtain $\gamma_1, \gamma_2, \ldots, \gamma_T$ and corresponding parameter vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T$ such that $\gamma_T \to \gamma^*$ and $f(\mathbf{x}; \mathbf{v}_i)$ approaches the degenerate distribution about $\mathbf{x}^*$. Let ρ be a real number between 0 and 1 representing the proportion of the sample taken as the elite sample. For a random sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ let $S_{(1)} \leq \ldots \leq S_{(N)}$ be the performances of $\{S(\mathbf{X}_i)\}$ ordered from smallest to largest. Thus, $S_{(j)}$ is the j-th *order-statistic* of the sequence $S(\mathbf{X}_1), \ldots, S(\mathbf{X}_N)$. γ_t is chosen to be the $(1-\rho)N$-th order statistic.

For fixed γ_t and $\mathbf{v}_{t-1}$, derive $\mathbf{v}_t$ from the solution of the following program

$$\max_{\mathbf{v}} D(\mathbf{v}) := \max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} I_{\{S(\mathbf{X}^{(i)}) \geq \gamma_t\}} \ln f(\mathbf{X}^{(i)}; \mathbf{v}) \tag{2}$$

The complete CE program is given in Algorithm 1.

Algorithm 1 CE Algorithm for Optimization

```
1. Choose an initial parameter vector $\mathbf{v}_0$. Set t = 1.
2. Generate a sample $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(N)}$ from the density $f(\cdot; \mathbf{v}_{t-1})$ and
   compute the sample $(1-\rho)$-quantile $\gamma_t$ of the performance according
   to $\gamma = S_{(\lceil (1-\rho)N \rceil)}$.
3. Using the same sample $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(N)}$, solve the stochastic program
   (2) and denote the solution by $\mathbf{v}_t$.
4. If for some $t \geq k$, say $k = 5$,
       $$\gamma_t = \gamma_{t-1} = \cdots = \gamma_{t-k}, \quad (3)$$
   then stop, otherwise set $t = t + 1$ and iterate from Step 2.
```

4 The Cross-Entropy Method for the Multiple Change Point Problem

Recall that d is the maximum number of change-points we wish to find. We can represent the position of the change-points as a non-decreasing d-dimensional vector. When the number of change-points is less than d the value of some components in the vector will be repeated, indicating the "same" change-point. We use a d-dimensional normal distribution, truncated to the integers 0 to L with independent components as our sampling distribution. We denote this distribution by $\tilde{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Let the mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$, $\mu_1 \leq \cdots \leq \mu_d$, and let the variance vector $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_d^2)$.

We choose initial values μ_i for the mean vector $\boldsymbol{\mu}$ such that each μ_i is equally spaced over the set $\{0, \ldots, L\}$ and $\boldsymbol{\sigma}$ is chosen appropriately large. That is, $\mu_i = \frac{i}{d}L$ and σ_i is of the order $\frac{L}{d}$. For each change-point vector in the sample we set $\hat{\theta}_n$ to the maximum likelihood estimator of θ_n, that is, the proportion of GC content in the segment defined by the change-points c_n and c_{n+1}. We wish to maximize the posterior probability (which is equivalent to the Maximum Likelihood estimate) given by:

$$S(\mathbf{c}) \propto \prod_{n=0}^{N} \hat{\theta}_n^{\mathbb{I}(c_n, c_{n+1})} (1 - \hat{\theta}_n)^{\mathbb{O}(c_n, c_{n+1})}. \quad (4)$$

The CE algorithm for the multiple change-point problem is as follows:

Algorithm 2 CE Algorithm for the multiple change-point problem

```
1. Choose initial values for $\boldsymbol{\mu}^0$ and $(\boldsymbol{\sigma}^2)^0$. Set $t = 0$.
2. Increase $t$ by 1. Generate a random sample $\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(N)}$ from the $\tilde{N}(\boldsymbol{\mu}^{t-1}, (\boldsymbol{\sigma}^2)^{t-1})$ distribution. That is, for all $j$, independently draw $C_j^{(i)}$ from the distribution $\tilde{N}(\mu_j^{t-1}, (\sigma^2)_j^{t-1})$.
3. For each $i = 1, \ldots, N$ order $C_1^{(i)}, \ldots, C_d^{(i)}$ from smallest to biggest, $C_1^{(i)} \leq \cdots \leq C_d^{(i)}$, and set $\mathbf{C}^{(i)} = (C_1^{(i)}, \ldots, C_d^{(i)})$.
4. Evaluate the performance of each $\mathbf{C}^{(1)} \ldots, \mathbf{C}^{(N)}$ using (4). Let $\mathcal{I}$ be the indices of the $N^{elite} = \rho N$ best performing samples.
5. For all $j = 1, \ldots, d$ let
       $$\tilde{\mu}_j^t = \sum_{i \in \mathcal{I}} C_j^{(i)} / N^{elite}$$
   and
       $$(\tilde{\sigma}_j^t)^2 = \sum_{i \in \mathcal{I}} (C_j^{(i)} - \tilde{\mu}_j^t)^2 / N^{elite}.$$
6. Smooth
       $$\boldsymbol{\mu}^t = \alpha \tilde{\boldsymbol{\mu}}^t + (1 - \alpha) \boldsymbol{\mu}^{t-1}, \quad (\boldsymbol{\sigma}^t)^2 = \alpha (\tilde{\boldsymbol{\sigma}}^t)^2 + (1 - \alpha)(\boldsymbol{\sigma}^{t-1})^2.$$
7. If $\max_j (\sigma_j^t)^2 < \varepsilon$, then go to step 8; otherwise, go to step 2.
8. For all $j = 1, \ldots, d - 1$ calculate
       $$\Delta_j = \mu_{j+1}^t - \mu_j^t.$$
9. Let $N = d - \#\{\Delta_j : \Delta_j < \delta\}$ be the number of change-points. Here $\delta$ is the minimum distance between two change-points for them to be considered different.
```

Algorithm 2 produces a single vector of change-points. A GC profile is a vector of length L where for each $i \in \{1, \dots, L\}$, $GC(i)$ is the average GC content in the segment containing the ith character in the sequence. A GC profile is produced from the change-point vector as follows:

$$GC(i) = \hat{\theta}_j \text{ where } C_{j-1} < i \leq C_j \text{ and } j = 1, \dots, d.$$

5 Results

In this section we look at the performance of the CE method with two examples. The first example is that of an artificially generated sequence from a known distribution. Using a known distribution allows direct comparison with existing methods in terms of the quality of the GC profile. The second example uses a real DNA sequence. We cannot know the true distribution for a real DNA sequence and therefore can only look for agreement in the GC profiles of the different methods.

5.1 Example 1: Artifical data

Let $(b_1, b_2, \dots, b_{22000})$ be a sequence of independent Bernoulli random variables generated with the parameters in Table 1.

Table 1: Bernoulli parameters for artificial sequence.

Positions	Bernoulli parameter
1 — 2000	$\theta_0 = 0.35$
2001 — 4000	$\theta_1 = 0.25$
4001 — 6000	$\theta_2 = 0.4$
6001 — 8000	$\theta_3 = 0.5$
8001 — 10000	$\theta_4 = 0.45$
10001 — 12000	$\theta_5 = 0.55$
12001 — 14000	$\theta_6 = 0.4$
14001 — 16000	$\theta_7 = 0.6$
16001 — 18000	$\theta_8 = 0.65$
18001 — 20000	$\theta_9 = 0.5$
20001 — 22000	$\theta_{10} = 0.4$

We generate 200 random sequences using these parameters and for each we run three algorithms:

- The MCMC approach in [6], taking 100 samples with a step size of 3000. This approach generates an average over the posterior distribution.
- The CE approach given in Algorithm 2 with a sample size of 1000, smoothing parameters of 0.7 for μ and 0.3 for σ and an elite proportion value ρ of 0.01.
- Taking the average of the GC profile produced over 10 runs of Algorithm 2 using a sample size of 100 and an elite proportion value ρ of 0.1. We denote this method the average CE (ACE) method. As the sample size is $\frac{1}{10}$th that of the CE method, and the average of 10 runs is taken, the running time should be approximately the same as the CE method.

To determine the quality of each method's profile, we calculate the Mean Square Error (MSE) given by MSE $= \sqrt{\sum_{i=1}^{22000}(t(i) - e(i))^2}$ where $t(i)$ is the value of the ith position of the GC profile produced using the parameters from Table 1, while $e(i)$ is the value at the ith position of the GC profile produced using the method to be tested. This is shown in the area between the two profiles in Figure 1.

The running times and average MSE distance values over the 200 random sequences are given in Table 2. All 200 sequences are generated with the same parameters and each method is run on each sequence once. This should ensure that, on average, a profile with a smaller MSE is actually more likely. If a single sequence was used 200 times it could be possible for the one profile to be more likely than another profile with a smaller MSE. It is clear that out of the two CE methods, taking the average of 10 runs each $\frac{1}{10}$th the size, on average, outperforms a single larger CE run. The profiles for the ACE method and the MCMC method from a single random sequence can be seen in Figure 2.

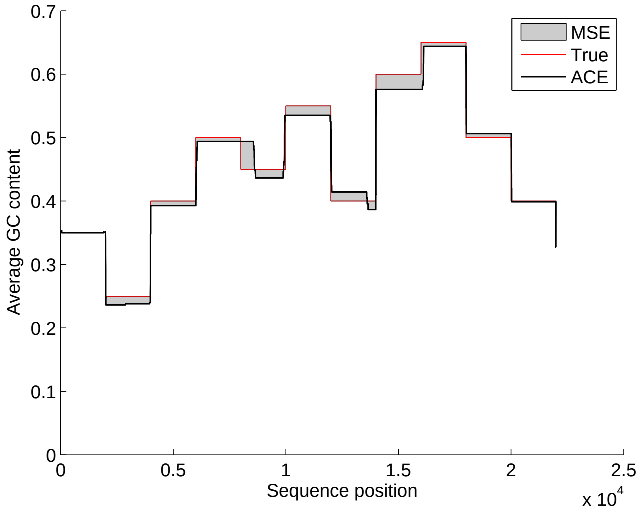

Figure 1: The error in a single ACE run when compared to the true distribution.

Table 2: The running time and average Mean Squared Error for the three different algorithms when applied to an artificial sequence of 22000 characters.

Algorithm	Time (sec)	MSE
MCMC	393	3.0
CE	18	3.4
ACE	18	3.0

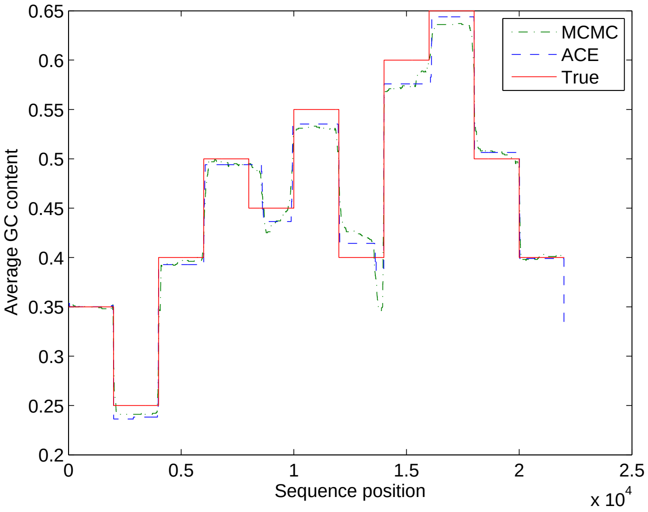

Figure 2: A profile plot comparing the average GC content along the full length of an artificial sequence as determined by the MCMC and ACE methods to the true profile.

Figure 2 shows that the ACE and MCMC methods are in excellent agreement with each other. Both identify the change-points very accurately, and when the calculated average GC content is higher or lower than the expected average GC content, they both differ in the same direction.

Since the average distance (MSE) is the same for the two methods, we look at the distribution of distances over the 200 sequences to see if there is any difference. Figure 3 shows the empirical distribution of distances for the ACE method and the MCMC method. Both methods have an average distance of about 3.0 but the ACE method has a greater number of simulation runs in the lower tail of the distribution. Figure 4 shows the density curves of the distances obtained from the three different methods.

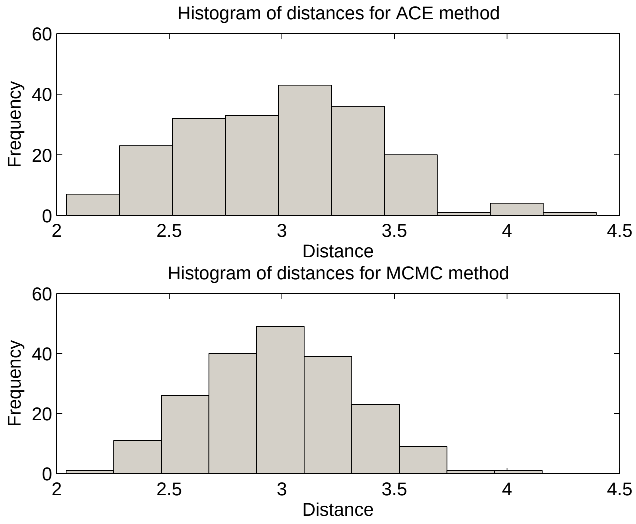

Figure 3: Histograms showing the empirical distribution of MSE distances from the true profile for the ACE method and MCMC method.

To directly compare the MCMC and ACE method we look at the difference in their two MSE distances for the same sequences. That is, for each of the 200 random sequences we calculate the value of $D_{\mathrm{ACE}}(B_i) - D_{\mathrm{MCMC}}(B_i)$, where B_i is the ith random sequence, D_{ACE} is the MSE distance from the ACE method and D_{MCMC} is the MSE distance from the MCMC method. The distribution of these differences is shown in Figure 5. A negative value indicates that the ACE method produced a profile closer to the true profile than the MCMC method, while a positive value indicates that the MCMC profile was closer to the true profile.

The density curve is centered around zero but it has a longer left tail. This shows that when looking at the same sequence, the two methods on average produce similar distances but the ACE method sometimes produces significantly better distances.

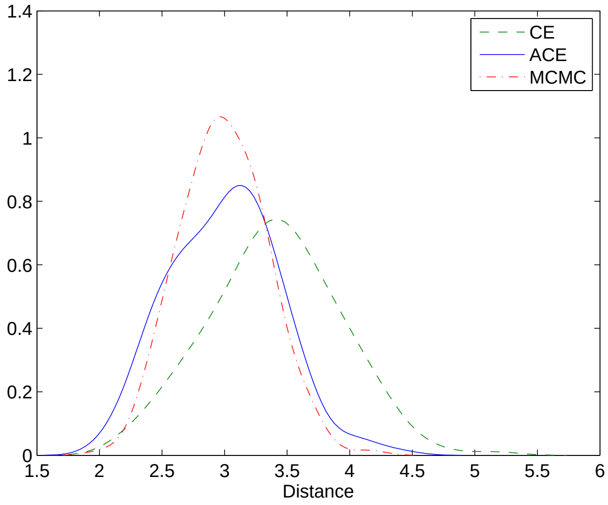

Figure 4: Density curves for the distribution of distances for the CE, ACE and MCMC methods. The density curves are obtained using MATLAB's kernel density estimation function `ksdensity()`.

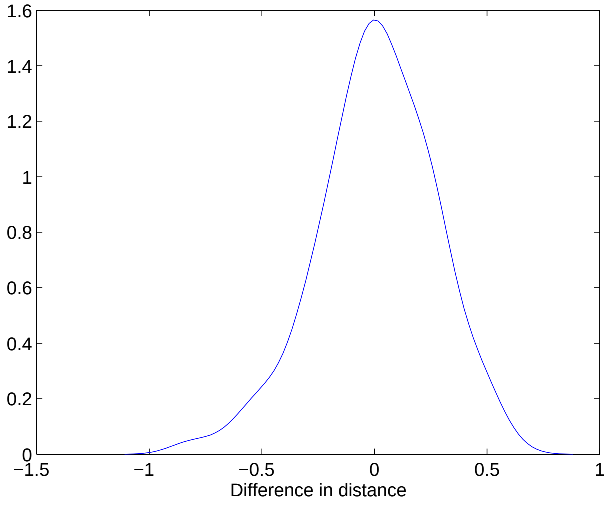

Figure 5: The density curve of the difference in distances of the ACE method and the MCMC method each applied to the same 200 random sequences.

5.2 Example 2: Real data

The second example we consider uses a segment of DNA know as the Human Major Histocompatibility Region (MHC) [9]. Due to this being real DNA, we do not know the true profile; instead we look for agreement between the different methods.

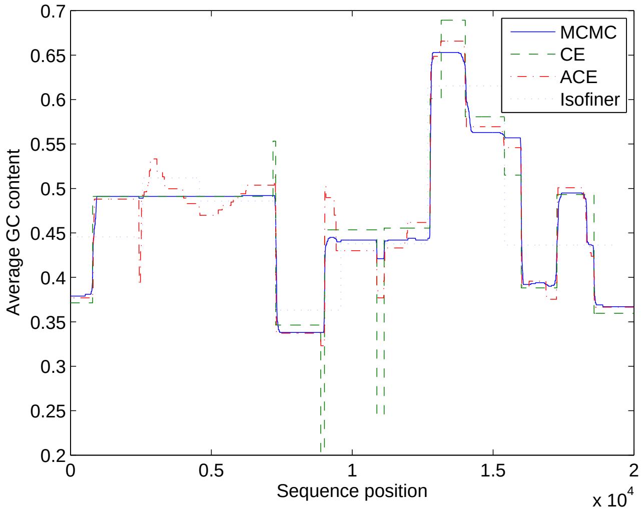

Figure 6: GC profiles for the ACE, CE and MCMC and IsoFinder methods on the MHC sequence. The two CE methods were run with the maximum number of change-points set to 20.

Figure 6 shows the GC profiles for the CE, ACE, MCMC and IsoFinder methods. It is clear that the four different methods all identify the major regions within the MHC sequence. IsoFinder identifies seven major regions while the other three methods all identify several smaller regions within these major regions. The MCMC, ACE and CE methods show excellent agreement in the identification of these smaller regions (< 2000 characters). The agreement between these methods allows for a great deal of confidence in the accuracy of the two CE methods as both IsoFinder and the MCMC method are well established.

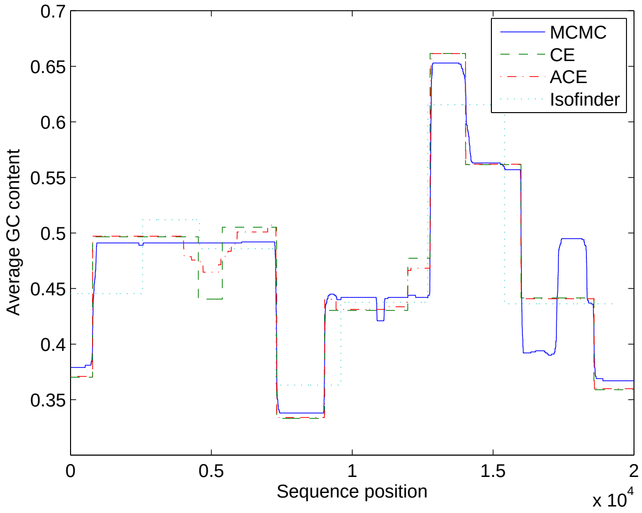

Figure 7: GC profiles for the ACE, CE and MCMC and IsoFinder methods on the MHC sequence. The two CE methods were run with the maximum number of change-points set to 10.

Figure 7 shows the same methods with the maximum number of change-points set to 10 for the two CE methods. From this figure it can be seen that the CE methods are now closer to the IsoFinder method than they are to the MCMC method. This is due to the fact that IsoFinder is only identifying the larger regions and when the maximum number of change-points is decreased for the CE methods they will identify fewer regions. The larger regions are more likely to have a greater impact on the likelihood of the profile when compared to the smaller regions and this is why they are still identified with a smaller maximum number of change-points.

6 Conclusion

Two new methods based on the CE approach have been proposed for the identification of change-points in DNA sequences. These methods have been shown to be highly effective on both artificial and real DNA sequences and compare well to existing techniques. The proposed methods have a clear speed advantage over existing MCMC methods while providing at least equally as good estimates. The proposed methods have the advantage of being able to identify a greater number of small regions compared to IsoFinder. While the two CE methods can identify many small regions, they are also capable of just identifing the larger regions by decreasing the maximum number of change-points.

7 Acknowledgments

G. Yu. Sofronov and D. P. Kroese acknowledge the support of an Australian Research Council discovery grant (DP0556631). D. P. Kroese acknowledge the support of an Australian Research Council discovery grant (DP0558957). J. M. Keith would like to acknowledge the support of Australian Research Council discovery grants (DP0879308, DP0556631) and a National Medical and Health Research Council grant "Statistical methods and algorithms for analysis of high-throughput genetics and genomics platforms" (389892).

References

[1] Isofinder website. `http://bioinfo2.ugr.es/IsoF/isofinder.html`.

[2] J. V. Braun and H.-G. Müller. Statistical methods for DNA sequence segmentation. *Statistical Science*, 13:142–162, 1998.

[3] Mouse Genome Sequencing Consortium. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.

[4] J. Keith, D. P. Kroese, and D. Bryant. A generalized Markov sampler. *Methodology and Computing in Applied Probability*, 6(1):29–53, 2004.

[5] J. M. Keith. Segmenting eukaryotic genomes with the generalized Gibbs sampler. *Journal of Computational Biology*, 13(7):1369–1383, 2006.

[6] J. M. Keith, P. Adams, S. Stephen, and J. S. Mattick. Delineating slowly and rapidly evolving fractions of the Drosophila genome. *Journal of Computational Biology*, 15(4):407–430, 2008.

[7] J. S. Mattick. The functional genomics of noncoding RNA. *Science*, 309(5740):1527 – 1528, September 2005.

[8] R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning.* Springer-Verlag, 2004.

[9] The MHC sequencing consortium. Complete sequence and gene map of a human major histocompatibility complex. *Nature*, (401):921–923, October 1999.

[10] G. Yu. Sofronov, G. E. Evans, J. M. Keith, and D. P. Kroese. Identifying change-points in biological sequences via sequential importance sampling. *Environmental Modeling & Assessment*, 2008. `http://www.springerlink.com/content/r1pk8657540647w8/fulltext.pdf`.