Mini Review

Investigating genomic structure using changept: A Bayesian segmentation model

Manjula Algama, Jonathan M. Keith *

School of Mathematical Sciences, Monash University, Clayton, VIC 3800, Australia

Abstract

Available online 27 August 2014

Keywords:
Sequence segmentation
Bayesian modelling
Generalised Gibbs sampler
Conservation levels
GC content
Non-coding RNA

Genomes are composed of a wide variety of elements with distinct roles and characteristics. Some of these elements are well-characterised functional components such as protein-coding exons. Other elements play regulatory or structural roles, encode functional non-protein-coding RNAs, or perform some other function yet to be characterised. Still others may have no functional importance, though they may nevertheless be of interest to biologists. One technique for investigating the composition of genomes is to segment sequences into compositionally homogenous blocks. This technique, known as ‘sequence segmentation’ or ‘change-point analysis’, is used to identify patterns of variation across genomes such as GC-rich and GC-poor regions, coding and non-coding regions, slowly evolving and rapidly evolving regions and many other types of variation. In this mini-review we outline many of the genome segmentation methods currently available and then focus on a Bayesian DNA segmentation algorithm, with examples of its various applications.

© 2014 Algama and Keith. Published by Elsevier B.V. on behalf of the Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Contents

1. Role of genome segmentation ...................................................... 108
2. Segmentation methods .......................................................... 108
  2.1. Sliding window analysis ..................................................... 108
  2.2. Hidden Markov models ..................................................... 108
  2.3. Multiple change-point analysis ............................................. 108
  2.4. Recursive segmentation method ......................................... 109
  2.5. Other segmentation methods ............................................. 109
3. Changept analysis ............................................................... 109
4. Method ................................................................................. 110
  4.1. Transforming alignment ..................................................... 109
    4.1.1. Segmentation based on a single property of interest (e.g.: conservation level) ........................................ 109
    4.1.2. Segmentation based on multiple properties (e.g.: conservation level, GC content and transition/transversion ratio) ..................................................... 109
  4.2. Modelling ........................................................................ 110
  4.3. Sampling ........................................................................... 112
  4.4. Applications of changept .................................................... 112
    4.4.1. Investigate segmentation patterns of genomic regions .............................................................. 112
    4.4.2. Identify alternatively spliced exons ........................................ 113
    4.4.3. Predict transcription factor binding sites (TFBS) ........................................................................ 113
    4.4.4. Identify putative non-coding RNAs ....................................... 114
    4.4.5. Identify rapidly evolving genomic regions ........................................ 114
5. Summary ............................................................................... 114
Acknowledgements .................................................................. 114
References ............................................................................... 114

* Corresponding author. Tel.: +61 3 990 20890; fax: +61 3 9905 4403.
E-mail address: jonathan.keith@monash.edu (J.M. Keith).

http://dx.doi.org/10.1016/j.csbj.2014.08.003
2001-0370/© 2014 Algama and Keith. Published by Elsevier B.V. on behalf of the Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
1. Role of genome segmentation

Identifying the distinct components of the human and other genomes is a core task in current bioinformatics, and a necessary pre-requisite to a full understanding of the connections between genomes and phenotypes. Yet the annotation of complex eukaryotic genomes is still far from complete. Even the proportion of the genome that performs biological functions is still hotly debated, with estimates varying from 5% [1] to 80% [2]. Whatever the true figure may be, it is clear that a vast amount of the biology underlying the structure of genomes remains to be discovered. Bioinformatics has an important role to play in this endeavour, and one of its tasks is to identify segments of the genome representing elements that require annotation.

2. Segmentation methods

Several techniques have been developed to analyse variation in properties of interest across a genome and to provide clues to the nature of its components. In this article we review some of the most widely used segmentation methods and discuss the main ideas behind each technique.

2.1. Sliding window analysis

Although not technically a segmentation method, ‘sliding window analysis’ is the most commonly used way to profile variation in a property of interest across a genome. This technique involves averaging the property of interest over a sliding window of a predetermined length along the sequence. For example if the window size is 10, the first point is obtained by averaging the property of interest over nucleotides 1–10, the second point is the average over nucleotides 2–11, and so on. Determining the window size can be crucial: a smaller window allows for a more precise localisation of changes, however this can increase the noise. Tajima in 1991 has proposed an algorithm to determine window size [3]. The main drawback of the sliding window analysis is that it does not identify boundaries where statistically significant changes to the property in question occur. To avoid some of the disadvantages of the sliding window approach, a windowless technique based on the Z curve was introduced to analyse GC content of genomic sequences [4]. This method enables calculation of GC content at any resolution, even at a base position. Some applications of the sliding window analysis can be found in papers [5–16].

2.2. Hidden Markov models

More precise segmentation methods have been developed to identify homogenous segments as well as the locations (change-points) at which sharp changes in a particular property of interest occurs. Hidden Markov models (HMMs) are one approach capable of inferring segment boundaries. The HMM methodology is well-established, dating from the 1950s [17]. In these models, the observed sequence is considered to be composed of segments, with the sequence of each segment generated by a Markov process. The transition probabilities for each segment are determined by a hidden state, and transitions between hidden states occur at segment boundaries. The sequence of hidden states is also modelled as a Markov process. A key parameter of an HMM is the order of the Markov chain, that is, the number of preceding sequence positions required to condition the transition probabilities of the observed sequence. This is unknown a priori, and usually needs to be specified, although some approaches are able to infer the order, or determine it adaptively.

HMMs were first used in biological sequence analysis by Churchill [18,19]. The parameters of the model, including segment boundaries, were estimated by using the maximum likelihood method based on the expectation–maximisation (EM) algorithm [20]. HMMs have since been widely used for sequence analysis problems in bioinformatics, and an extensive literature now exists. Two important developments were the 1998 GeneMark.hmm algorithm which used an HMM to find exact gene boundaries [21] and an HMM developed by Peshkin and Gelfand in 1999 to segment yeast DNA sequences [22]. Some other important examples are included in [23–29]. The Sarmen package of Python modules built by Gueguen for easy building and manipulation of sequence segmentations uses both sliding window and HMM methods [30].

HMM models have also been implemented from a Bayesian perspective. One advantage of adopting a Bayesian approach is that it provides quantification of the uncertainties in parameter estimates in the form of probability distributions. In fact, one can dispense with point estimates of parameters altogether, instead reporting marginal distributions for key parameters, such as the locations of change-points. Boys et al. in 2000 presented a Bayesian method of segmentation using HMMs when the number of segments is known [31] and later generalised this method for an unknown number of segments [32]. In 2006, the segmentation method developed by Kedzierska and Husmeier was a combination of the sliding window analysis and the Bayesian HMM [33]. Nur and co-workers in 2009 performed sensitivity analysis on priors used in the Bayesian HMM to show the impact of prior choice on posterior inference [34]. One challenge for Bayesian HMM approaches is that they are computationally intensive and are typically infeasible for segmenting large-scale sequences, without simplifying heuristics.

2.3. Multiple change-point analysis

This approach arose independently of HMMs, and has an extensive literature dating back to the 1970s [35,36]. Change-point analysis differs from HMMs in that it typically assumes no Markov dependence in either the observed sequence or the underlying sequence of hidden states. In this sense change-point models are simpler than HMMs, and have fewer parameters. However, the two types of analysis are clearly related, and it may be useful to think of change-point models as zeroth order HMMs. A key advantage of change-point models, due to their simplicity, is that they are reduced computational burden, a point which is of particular relevance when implementing them within a Bayesian framework.

The use of multiple change-point models in bioinformatics was pioneered by Liu and Lawrence in 1999, using a Bayesian framework [37]. In 2000, Ramsenks et al. developed a similar method which uses a Bayesian estimator to measure the degree of homogeneity in segmentation [38]. In this method, optimal segmentation is obtained by maximising the likelihood function using the dynamic programming technique presented in [39]. After completion, the partition function approach is used to obtain segmentation with longer segments by filtering the boundaries. In contrast to the approach of Liu and Lawrence, this method does not use probability distributions for segment boundaries and does not use sampling. A related method is presented in [40], which uses reversible jump Markov chain Monte Carlo (RJMCMC) sampling method to estimate posterior probabilities [41]. In contrast to Liu and Lawrence, they have used Poison intensity models as the underlying model (as opposed to multinomial likelihood). The method has been tested by applying to modelling the occurrence of ORFs along the human genome. Another Bayesian model can be found in [42].

The method on which we focus in the main part of this article [43,44] is also of this type. The method can be described as a segmentation–classification model as it not only detects change-points but also groups segments based on their sequence characteristics. The group to which a segment belongs is essentially a hidden state, in the terminology of HMMs, and the classification is unsupervised, in the terminology of machine learning. There are two main innovations in this method. The first is that the character frequencies (emission probabilities) for a given segment are not constant for all segments in a group. Instead, the character frequencies are drawn from a Dirichlet distribution specific to the group to which that segment belongs, and it is the
parameters of this distribution that characterise the group. There is thus an additional layer to this hierarchical model, and this layer is another characteristic distinguishing the model from HMMs. Allowing variation in the character frequencies for segments in a group means that this model can be used to dissect multi-modal distributions of properties of interest, a central feature in recent applications [45,46]. The second innovation in this method is the use of the Generalised Gibbs Sampler (GGS) [47], a new technique in Markov chain Monte Carlo simulation. The GGS provides highly efficient sampling from a varying dimensional space (important here as the number of change points is variable).

2.4. Recursive segmentation method

The recursive segmentation method finds segment boundaries that maximise the difference in base compositions between adjacent segments with respect to some predefined compositional measure (Jensen–Shannon divergence — $D_{JS}$). The process is repeated until further segmentation of sequence segments produces no statistically significant improvements. The recursive segmentation method has been widely applied to segmentation problems such as isochore detection or detection of CpG islands [48–52]. More recent applications include locating borders between coding and non-coding regions of bacteria genomes [53] and in developing IsoPlotter: a tool for studying the compositional architecture of genomes [54].

The recursive segmentation method presented in [55] is significant in that it does not require specification of the number of segment classes (something most of the other methods require). This method has been successfully used to identify alien DNAs in bacterial genomes, detect structural variants in cancer cell lines and perform alignment-free genome comparisons.

2.5. Other segmentation methods

Methods based on least squares estimation [56] and wavelet analysis [57] have also been used. Sequential importance sampling (SIS) [58], the cross-entropy method [59] and the Bayesian adaptive independence sampler [60] have also been used to find segment boundaries and parameters of the process in each segment.

Olshen et al. developed the circular binary segmentation method (CBS) in 2004 for the analysis of array-based comparative genomic hybridisation (array-CGH) data [61]. CGH (comparative genomic hybridization) is a technique for measuring DNA copy numbers at thousands of locations on a genome. The modification of conventional CGH to obtain high resolution data is called array-CGH. The variation in DNA copy number is often used to identify cancer progression. The CBS algorithm divides the genome into regions of equal DNA copy number and identifies the genomic locations of copy number transitions (change-points). In 2007, changes were made to the original CBS algorithm to enhance the speed by introducing, (1) a hybrid approach for the computation of the $p$-value and (2) a stopping rule for early identification of change-points [62].

In 1996, Tibshirani proposed a new method called ‘lasso’ (least absolute shrinkage and selection operator) for estimation in regression models, which involves constraining the sum of the absolute values of the regression coefficients [63]. This produces some coefficients that are exactly zero and hence gives interpretable models. In 2006 ‘fused lasso’ – a generalisation of ‘lasso’ – was introduced to handle problems with features that can be ordered in some meaningful way [64]. The fused lasso penalises the sum of the absolute values of the coefficients and their successive differences. The method was applied along with the CBS method to estimate the copy number alterations in breast tumour data (CGH data of breast cancer cell line MDA157) [65]. CBS had difficulties in detecting change points whose alteration signals are weak (chromosome 7 and 15 of the selected cell line), but the fused lasso successfully recognised various copy number alterations. Besides identifying gains and losses in CGH data, the fused lasso can also be generalised to other analysis; for example, understanding the interactions between copy number alternations and mRNA expression levels.

Determining the number of change-points is an important aspect of change-point analysis. In 2007, Zhang et al. proposed the modified Bayes Information Criterion (BIC) as a model selection procedure for array-CGH data analysis [66]. The first term of the modified BIC is similar to the classic BIC (consisting of the log likelihood), but it differs in the terms that penalise for model dimension. One of the advantages of using the modified BIC is that it does not require a specific prior or tuning parameters, but it can only be applied to normally distributed, uncorrelated and homoscedastic data. However the modified BIC is not limited to the analysis of array-CGH data. Some other methods that adaptively determine the number of change-points can be found in [41,46,67].

The multi-scale segmentation method developed by Futschik and co-workers also estimates the number of segments and their boundaries simultaneously [68]. One advantage of this method is that it does not require distributional assumptions regarding the lengths of segments. Another feature is that this method is able to choose an appropriate number of segments with user specified probability $1 - \alpha$.

Many early statistical segmentation methods were reviewed in [69]. Elhaik et al. reviewed the performance of seven recent algorithms by segmenting human chromosome 1 based on variability of GC content [70].

3. Changept analysis

In the remainder of this mini-review, we focus on the changept program developed by Keith et al. [43,44]. This is a Bayesian multiple change-point algorithm capable of simultaneously segmenting a genomic alignment and classifying segments into one of predefined number of segment classes. Segments can be classified according to multiple properties including level of evolutionary conservation between species, GC content and transition/transversion ratio. Program readcp is a part of the changept package that takes the outputs produced by changept and estimates, for each genomic position, the probability that genomic position belongs to each segment class. The package uses a highly efficient sampling technique known as the Generalised Gibbs Sampler [47] resulting in a highly efficient algorithm that enables chromosome or even genome-wide analysis. The algorithm can be used to segment a genomic alignment based on a single property of interest or multiple properties. There is no limit on number of aligned species.

4. Method

4.1. Transforming alignment

4.1.1. Segmentation based on a single property of interest (e.g.: conservation level)

Suppose we want to segment a pairwise alignment of size $L$ based on the degree of conservation between two species. The first step is to convert the alignment into a binary sequence by replacing the alignment columns in which two DNA sequences match with a ‘1’ and replacing columns in which they mismatch with a ‘0’. The gaps between alignment blocks are marked by a ‘#’ symbol and these are considered as fixed change-points by the model. The indels (alignment gaps) in the reference species are not encoded while indels in other species are encoded using letter ‘I’ which will be excluded from the final analysis of the sequence. The binary sequence generated in this way is used as the input for the program changept.

4.1.2. Segmentation based on multiple properties (e.g.: conservation level, GC content and transition/transversion ratio)

In segmenting a pairwise alignment based on more than one property of interest, one possibility is to use a 16-character representation
4.2. Modelling

The complete model is presented in [43,44]. Here we only present the main idea behind the model. The process of Bayesian modelling consists of 3 main steps [72]: (1) set up a joint probability distribution for all the variables in a problem; (2) calculate posterior distribution — the conditional probability distribution of the unobserved parameters of interest, given the observed data; (3) evaluate the model. Step (1) starts with writing down the likelihood function of the model, i.e. probability of the observed quantities given unknown parameters. This describes the stochastic process by which sequences are generated, and consequently it quantifies the probability of generating the observed sequence for any given parameter values.

In writing down the likelihood function of our model, we denote the probability of starting a new segment by \( \phi \), the number of fixed change-points by \( k' \) and the total number of change-points (including fixed change-points) by \( k \). The positions of change-points are denoted by \( C = (c_1, c_2, \ldots, c_k) \). We set \( c_0 = 1 \). For each position in the sequence, except for the first position and those immediately following a fixed change-point (marked by ‘#’), a decision has to be made whether to start a new segment. Thus the probability of generating a segmentation with \( k \) change-points at \( C = (c_1, c_2, \ldots, c_k) \) positions is given by:

\[
p(k, C|\phi) = \phi^{k'} (1-\phi)^{k-k'}
\]

where \( L \) is the length of the sequence \( S \).

Each segment is then assigned to one of \( \omega \) conservation classes. Let \( \pi_i \) denotes the probability of assigning a segment to class \( t \). We denote the class to which segment \( i \) is assigned by \( g_i \in \{0, 1, \ldots, \omega - 1\} \) and let \( g = (g_0, g_1, \ldots, g_k) \). The probability that \( x_0 \) segments are assigned to class \( 0 \), \( x_1 \) segments are assigned to class \( 1 \), ..., \( x_{\omega-1} \) segments are assigned to class \( \omega - 1 \) is:

\[
p(g|k, \pi) = \prod_{i=0}^{k} \pi_{g_i}^{x_{g_i}}
\]

In the case of the binary representation of the sequence \( S \), let \( \theta_i \) represent the probability of generating a ‘1’ in each position of segment \( i \) in class \( t \). Each \( \theta_i \) is independently drawn from the following beta distribution with unknown parameters \( \alpha_0(i) \) and \( \alpha_1(i) \):

\[
p(\theta_i | \alpha_0(i), \alpha_1(i)) = \frac{\Gamma(\alpha_0(i) + \alpha_1(i))}{\Gamma(\alpha_0(i)) \Gamma(\alpha_1(i))} \theta_i^{\alpha_0(i)-1} (1-\theta_i)^{\alpha_1(i)-1}.
\]

Here \( \theta = (\theta_0, \theta_1, \ldots, \theta_k) \), \( \alpha^{(i)} = (\alpha_0(i), \alpha_1(i)) \) and \( \alpha = (\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\omega-1)}) \).

This can be generalised when \( S \) represents the alignment formed using a finite alphabet \( \{1, \ldots, D\} \) (D-character representation). Let \( \theta_{ij} \) represent the probability of generating character \( j \) in segment \( i = 0, \ldots, k \). We denote \( \theta = (\theta_{11}, \ldots, \theta_{kD}) \). Then for each segment \( i \) in class

\[
\begin{align*}
\text{Species 1} & \quad A \quad A \quad A \quad A \quad A \quad A \\
\text{Species 2} & \quad A \quad C \quad G \quad T \quad A \quad C \\
\text{Species 3} & \quad A \quad C \quad G \quad T \quad A \quad C \\
\text{Symbol} & \quad a \quad b \quad c \quad d \quad e \quad f \quad g \quad h \quad i \quad j \quad k \quad l \quad m \quad n \quad o \quad p
\end{align*}
\]
gi, θi are drawn from a Dirichlet distribution \( p(\theta | g_i, \alpha) \) with parameter vector \( \alpha = (\alpha_1^g, …, \alpha_D^g) \) for each class.

The binary sequence within each segment \( i \) is generated by independent Bernoulli trials at each position in the segment. Thus the probability that segment \( i \) contains specific sequence \( S_i \) including \( m_i \) number of '0's and \( n_i \) number of '1's is given by:

\[
p(S_i | L_i, \theta_i) = \theta_i^{m_i} (1- \theta_i)^{n_i}
\]

where \( L_i = c_i + 1 - c_i \) is the length of segment \( i \).

In using the D-character representation, we assume that within each segment, the sequence is generated by independent trials with \( D \) possible outcomes. Let \( m_{ij} \) be the number of times character \( j \) appears in segment \( i \). Thus the likelihood of an observed DNA sequence can be written as:

\[
p(S | k, c, \pi, \alpha) = \prod_{i=0}^{k} \prod_{j=1}^{D} \theta_i^{m_{ij}}.
\]

The final sequence is obtained by concatenating sequences \( S_0, …, S_k \). Therefore the joint distribution of parameters \( k, c, \theta \) and \( S \) is given by:

\[
(k, c, g, \theta, S, \phi, \pi, \alpha) = p(k, c, \phi, \pi, \alpha) \prod_{i=0}^{k} B \left( \theta_i | \alpha^{(g)} \right) p(S_i | L_i, \theta_i).
\]

The prior probabilities assigned to parameters \( \phi, \pi \) and \( \alpha \) are given in [44]. Using Bayes theorem, integrating over \( \phi \) and \( \theta \), and summing over \( g \), the following posterior distribution is obtained:

\[
p(k, c, \pi, \alpha | S) = \Gamma(L-k) \Gamma(k-k') \prod_{i=0}^{k} f(m_i, n_i | \pi, \alpha)
\]

where

\[
f(m, n | \pi, \alpha) = \sum_{\pi} \frac{\Gamma(m+\alpha_0^{(g)}+\alpha_1^{(g)}) \Gamma(n+\alpha_0^{(g)}+\alpha_1^{(g)})}{\Gamma(\alpha_0^{(g)}) \Gamma(\alpha_1^{(g)})}.
\]

---

### Table 3

8-character representation used to encode a pairwise alignment.

| Symbol | a | b | c | d | e | f | g | h |
|--------|---|---|---|---|---|---|---|---|
| Species 1 | A | T | A | T | A | T | C | G |
| Species 2 | A | T | G | G | C | T | A | A |

---

**Fig. 2.** The changept workflow. This figure illustrates the sequence of steps generally followed in analysing a set of DNA sequences by using the program changept. In step 3, \( T \) represents the number of segment classes specified by the user.
obtain the posterior distribution of parameters updates the change-point positions. These different types of updates segment and either inserts a change-point, deletes a change-point or alternative to the reversible-jump sampler [41]. It cycles through each that the number of change-points is varying and thus provides an alignment.

3 sequence (coding 1) alignment, C) 16-class model for the Segmentation characteristics of two genomic regions. Table 4; Fig. 3.

\[ p(k, C, \Theta, \phi, \alpha, g, \pi|S) \propto p(\phi)p(k, C|\phi)p(\alpha)p(\pi) p(g|k, \pi) \prod_{i=1}^{k} p(\theta|\alpha, g, \pi)p(S|k, C, \Theta). \]

Here \( p(\phi), p(\alpha) \) and \( p(\pi) \) denote the prior probabilities assigned to parameters \( \phi, \alpha \) and \( \pi \) [43]. In simplifying further, it is possible to integrate the above equation over \( \phi \) and \( \Theta \) and to take sum over \( g \) to obtain the posterior distribution of \( p(k, C, \alpha, \pi|S) \).

Fig. 1 shows the parameter dependencies of the model.

4.3. Sampling

The posterior distribution is sampled using the Generalised Gibbs Sampler (GGS), a Markov chain Monte Carlo technique [47]. Unlike the conventional Gibbs sampler, the GGS takes into account the fact that the number of change-points is varying and thus provides an alternative to the reversible-jump sampler [41]. It cycles through each segment and either inserts a change-point, deletes a change-point or updates the change-point positions. These different types of updates are referred as ‘move-types’ which are analogous to the coordinate updates of the conventional Gibbs sampler.

Once the alignment is transformed into the changept input sequence, it is then run through the program changept (source code is available upon request) to produce a user specified number of samples.

The next step of changept analysis is to check if convergence to the limiting distribution has occurred. This is most commonly assessed by inspecting a time-series plot of the log-likelihood against the sample number. The same plot is used to decide the length of the ‘burn-in’ period. Changept currently requires the user to specify the number of segment classes \((T)\). Selecting the model with the most appropriate number of classes can be done by using either of the following methods: (1) investigating AIC, BIC and DICV plots [67]; and (2) investigating the stability of each segment class [46]. The final model is then run through the program readcp to calculate profile values. The profile shows the probability that each position in the input sequence belongs to one of the segment classes in the selected model. These posterior probabilities are estimated using Monte Carlo integration. These outputs (a profile file for each segment class in the final model) are used to generate WIG/BED files that can be uploaded to a genome browser (e.g. http://genome.ucsc.edu/) for viewing gene-related information.

This workflow is illustrated in Fig. 2 and a full description of how to use changept and readcp can be found in [73].

4.4. Applications of changept

In this section we discuss several applications of program changept. These can be categorised into sub-headings:

- Investigate segmentation patterns of genomic regions
- Identify alternatively spliced exons
- Identify putative transcription factor binding sites (TFBS)
- Identify putative non-coding RNAs
- Identify rapidly evolving genomic regions.

In each sub-heading we provide examples to illustrate the performance of the program changept.

4.4.1. Investigate segmentation patterns of genomic regions

This section summarises the results of [46]. The program changept was applied to three possible pairwise alignments of 3’UTR among three closely related Drosophila species: Drosophila melanogaster, Drosophila simulans and Drosophila yakuba. We also segmented three randomly selected portions of the alignment of D. melanogaster to D. simulans protein-coding sequences of the same length as the 3’UTR alignment of that pair. This was required as the number of segment classes detectable is sensitive to the length of the changept input sequence. These alignments were obtained from http://genomics.princeton.edu/AndolfattoLab/Andolfatto_Lab.html. Each pairwise alignment is encoded using an 8-character representation (Table 3) that

| Alignment          | Component   | Model       | No. of alignment columns | No. of fixed change-points | Posterior average no. of change-points | Posterior average length of segments |
|--------------------|-------------|-------------|--------------------------|---------------------------|----------------------------------------|--------------------------------------|
| Dme vs Dsi         | 3’UTR       | 15          | 2,678,635                | 9112                      | 50,001                                 | 54                                   |
| Dme vs Dya         | 3’UTR       | 16          | 2,486,711                | 8622                      | 53,051                                 | 49                                   |
| Dsi vs Dya         | 3’UTR       | 15          | 2,481,568                | 8607                      | 51,547                                 | 48                                   |
| Dme vs Dsi Coding  | Coding 1    | 12          | 2,680,987                | 6760                      | 11,086                                 | 242                                  |
| Dme vs Dsi Coding  | Coding 2    | 12          | 2,681,121                | 6626                      | 10,190                                 | 263                                  |
| Dme vs Dsi Coding  | Coding 3    | 14          | 2,681,284                | 6463                      | 9982                                   | 268                                  |

\[ ^{a} \text{Dme: } D. \text{ melanogaster.} \]
\[ ^{b} \text{Dsi: } D. \text{ simulans.} \]
\[ ^{c} \text{Dya: } D. \text{ yakuba.} \]
\[ ^{d} \text{Coding 1, 2, 3: three different randomly selected protein-coding sequences.} \]
captures degree of conservation between two species, GC content and transition/transition ratio.

In order to select the optimal number of segment classes for each alignment, we performed separate segmentation analysis using models with 1–20 segment classes \((T = 1,.., 20)\). After assessing stability of segment classes in each model of 3'UTRs, we selected the 15-class model for the \(D. melanogaster\) and \(D. simulans\) alignment, the 16-class model for the \(D. melanogaster\) versus \(D. yakuba\) alignment and the 15-class model for the \(D. simulans\) versus \(D. yakuba\) alignment. Further we selected the 12-class model for the \(D. melanogaster\) versus \(D. simulans\) two protein-coding sequences (coding 1 and coding 2) and the 14-class model for the third protein-coding sequence, and consequently the average segment length in 3'UTRs is about one fifth of that in the coding sequence. This evidence suggests that \(Drosophila\) 3'UTRs contain more numerous sub-units than protein-coding sequences.

4.4.2. Identify alternatively spliced exons

This example was extracted from work presented by Boyd SE and co-workers in segmenting a 3-way alignment (human, mouse and rat DNA sequences) of the GFAP gene [74]. Fig. 4 shows a section of the WIG file (uploaded to the UCSC genome browser) of the segment class that corresponds to regions of high conservation among human, mouse and rat of the GFAP gene. In general, the start and end points of the conserved features occur at or very close to the boundaries of the exons (e.g. exon 6 in right of the screen). In the case of exons 7 and 7a (as labelled), the conserved features do not terminate immediately after the end of the annotated exon boundaries. The conserved feature corresponding to exon 7 extends for 30 nucleotides into intron 7 and the feature corresponding to exon 7a begins 50 nucleotides upstream of the start of exon 7a.

To find the possible novel splicing sites associated with exon 7a, the human DNA sequence of the extended region has been submitted to the Human Splicing Finder server (http://www.umd.be/HSF/HSF.html). The HSF predicts a potential acceptor splice site located 40 nt upstream of the conserved region (marked by HSF2 in Fig. 4), supporting the hypothesis of a new splice variant of the GFAP gene.

4.4.3. Predict transcription factor binding sites (TFBS)

Identifying putative TFBS is yet another interesting application of the program changept. To test this, we selected the pairwise alignment (human versus mouse) of the SHH gene which contains experimentally identified regulatory elements within the upstream regulatory region [75]. We used LAGAN (http://lagan.stanford.edu/lagan_web/index.shtml) [76] to align the two DNA sequences. The alignment was encoded using the 16-character representation. Based on the
investigation of DICV values, the 6-class model was selected for human and mouse 2-way alignment. Interestingly, for SHH, the positions of annotated exons were not identified as belonging to the most conserved segment class (90% conservation), rather they were identified to belong to the second most conserved class (85% conservation). Fig. 5 depicts the WIG profiles of these two most conserved segment classes.

Features A and B (Fig. 6) are regions identified as belonging to the most conserved class. These regions have been experimentally identified as regulatory elements [75]. This result confirms that regions predicted by changept (features A and B) are in appropriate locations for transcription factor binding. We are currently investigating the potential of changept for genome-wide detection of TFBS.

4.4.4. Identify putative non-coding RNAs

Non-coding RNA (ncRNA) is an RNA molecule that is not translated into a protein. It has been estimated that 98% of human genomic output is ncRNAs, however what proportion of ncRNAs are functional and the functions of many ncRNAs remain unknown [77]. The program changept can be used to identify highly conserved non-coding regions in genomes that are likely to be functional. To provide an example, we can use the WIG profiles of the two most conserved segment classes of SHH gene (Fig. 5). The top profile shows features that are even more conserved than the annotated protein-coding regions. Further, changept has predicted conserved features in the 2nd most conserved class that are equally conserved as exons. These highly conserved elements could contain either ncRNAs or regulatory sequences. In a recent project, we are working with biologists to investigate these and other putative ncRNAs identified using changept in a number of genomes.

4.4.5. Identify rapidly evolving genomic regions

The work presented in [44] provides an example for this changept application. To summarise the main findings, program changept has been applied on three whole-genome and three partial-genome pairwise alignments of eight Drosophila species. Three main classes of conservation level have been identified, comprising slowly evolving, rapidly evolving and intermediate segments. In a recent project, we are applying changept to three malaria species to identify genomic regions likely to be involved in the ability of the malaria parasite to infect their host species.

5. Summary

In this mini-review, we discussed various algorithms that can be used to segment genomic sequences. We also outlined the mathematics and methods of program changept, a Bayesian segmentation algorithm that is capable of segmenting an alignment while simultaneously classifying segments into different segment classes that share similar properties. We have demonstrated the effectiveness of this method through examples. The program changept can be used to identify putative functional elements in genomes such as non-coding RNAs, alternatively spliced exons and transcription factor binding sites. Other applications of program changept include identifying rapidly evolving genomic regions and inferring various segmentation patterns in genomic regions.

Acknowledgements

We would like to thank Dr. Robert Bryson-Richardson and Edward Tasker for their collaboration on changept applications. This work was supported by the Australian Research Council (grant DP1095849).

References

[1] Waterston RH, Lindblad-Toh K, Birney E, Rogers J, Abril JF, Agarwol P, et al. Initial sequencing and comparative analysis of the mouse genome. Nature 2002;420:520–62.
[2] Bernstein BE, Birney E, Dunham I, Green ED, Gunter C, Snyder M. An integrated encyclopedia of DNA elements in the human genome. Nature 2012;489:57–74.
[3] Tajima F. Determination of window size for analyzing DNA sequences. J Mol Evol 1993;33:470–3.
[4] Zhang CT, Wang J, Zhang R. A novel method to calculate the G + C content of genomic DNA sequences. J Biomol Struct Dyn 2001;19:333–41.
[5] Bernardi G. Misunderstandings about isochores. Part 1. Gene 2001;276:3–13.
[6] Clay O, Caresi N, Douday C, Macaya G, Bernardi G. Compositional heterogeneity within and among isochores in mammalian genomes. I. CCG and sequence analyses. Gene 2001;276:15–24.
[7] Lande R, Linton LM, Birren B, Nusbaum C, Zody MC, et al. Initial sequencing and analysis of the human genome. Nature 2001;409:860–921.
[8] Costantini M, Auleta F, Bernardi G. Isochore patterns and gene distributions in fish genomes. Genomics 2007;90:364–71.
[9] Costantini M, Clay O, Auleta F, Bernardi G. An isochore map of human chromosomes. Genome Res 2006;16:536–41.
[10] Lenhard B, Sandelin A, Mendoza L, Engstrom P, Jareborg N, Wasserman WW. Identification of conserved regulatory elements by comparative genome analysis. J Biol 2003;2:12.
[11] Turner TL, Hahn MW, Nuzhdin SV. Isochroine evolution in speciation in Anopheles gambiae. PLoS Biol 2005;3:e285.
[12] Spellman PT, Rubin GM. Evidence for large domains of similarly expressed genes in the Drosophila genome. J Biol 2002;1:5.
[13] Takami H, Nakaoke K, Takai Y, Maeno G, Sasaki R, Masai N. Complete genome sequence of the alkaliphilic bacterium Bacillus halodurans and genomic sequence comparison with Bacillus subtilis. Nucleic Acids Research 2000;28:4317–31.
[14] Karlin S. Detecting anomalous gene clusters and pathogenicity islands in diverse bacterial genomes. Trends Microbiol 2001;9:335–43.
[15] Fares MA, Elena SF, Ortiz J, Moya A, Barrio E. A sliding window-based method to detect selective constraints in protein-coding genes and its application to RNA viruses. J Mol Evol 2002;55:509–21.
[16] Carlson CS, Thomas DJ, Eberle MA, Swanson JE, Livingston RJ, Rieder MJ, et al. Genomic regions exhibiting positive selection identified from dense genotype data. Genome Res 2005;15:1553–65.
[17] Stratonovich R. Conditional Markov processes. Theory Probab Appl 1960;5:156–78.
[18] Churchill GA. Stochastic models for heterogeneous DNA sequences. Bull Math Biol 1989;51:79–94.
[19] Churchill GA. Hidden Markov chains and the analysis of genome structure. Comput Chem 1992;16:107–15.
[20] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B Methodol 1977;39:1–38.
[21] Lukashin AV, Borodovsky M. GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res 1998;26:1107–15.
[22] Peshkin L, Gelfand MS. Segmentation of yeast DNA using hidden Markov models. Bioinformatics 1999;15:980–6.
[23] Nicolas F, Bize L, Muri F, Hoebeke M, Rodolphe F, Ehrlich SO, et al. Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models. Nucleic Acids Res 2002;30:1418–26.
[24] Azad RK, Borodovsky M. Probabilistic methods of identifying genes in prokaryotic genomes: connections to the HMM theory. Brief Bioinform 2004;5:118–30.

Fig. 6. Conserved regions correspond to TFBS in SHH identified by program changept. The profile shows the conserved features predicted by program changept in the upstream of SHH gene, genomic coordinates — chr7:155,604,884–155,605,370. The locations of TFBS are marked by red arrows.
[25] Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. Hidden Markov models in computational biology: applications to protein modeling. J Mol Biol 1994;235:1501–31.

[26] Stjernqvist S, Ryder T, Skold M, Staaf J. Continuous-index hidden Markov modelling of array CGH copy number data. Bioinformatics 2007;23:1006–14.

[27] Marioni JC, Thorne NP, Tavare S. A heterogeneous hidden Markov model for segmenting array CGH data. Bioinformatics 2006;22:1144–6.

[28] Willerbrock H, Fridlyand J. A comparison study: applying segmentation to array CGH data for downstream analyses. Bioinformatics 2005;21:4084–91.

[29] Fridlyand J, Snijders AM, Pinkel D, Albertson DG, Jain AN. Hidden Markov models approach to the analysis of array CGH data. J Multivar Anal 2004;90:132–53.

[30] Gueguen L. Sarment: Python modules for HMM analysis and partitioning of sequences. Bioinformatics 2005;21:3427–8.

[31] Boys RJ, Henderson DA, Wilkinson DJ. Detecting homogeneous segments in DNA sequence segmentations. Bioinformatics 2002;18:226–32.

[32] Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. Hidden Markov models in computational biology: applications to protein modeling. J Mol Biol 1994;235:1501–31.

[33] Hawkins DM. Testing a sequence of observations for a shift in location. J Am Stat Assoc 1979;74:365.

[34] Nur D, Allingham D, Rousseau J, Mengersen KL, McVinish R. Bayesian hidden Markov models for segmenting array CGH data for downstream analyses. Bioinformatics 2009;25:4084–81.

[35] Hawkins DM. Testing a sequence of observations for a shift in location. J Am Stat Assoc 1979;74:365.

[36] Liu JS, Lawrence CE. Bayesian inference on biopolymer models. Bioinformatics 1999;15:38–49.

[37] Finkelstein AV, Roytberg MA. Computation of biopolymers: a general approach to the analysis of DNA sequences. Comput Chem 2002;26:491–509.

[38] Gueguen L. Sarment: Python modules for HMM analysis and partitioning of sequences. Bioinformatics 2005;21:3427–8.

[39] Futschik A, Hotz T, Munk A, Sieling H. Multiscale DNA partitioning: statistical evidence for segments. Bioinformatics 2008;24:2184–91.

[40] Boys RJ, Henderson DA, Wilkinson DJ. Detecting homogeneous segments in DNA sequence segmentations. Bioinformatics 2002;18:226–32.

[41] Green PJ. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 1995;82:711–32.

[42] Husmeier D, Wright F. A Bayesian approach to discriminate between alternative DNA sequence segmentations. Bioinformatics 2002;18:226–34.

[43] Keith JM. Segmenting eukaryotic genomes with the Generalized Gibbs Sampler. J Comput Biol 2006;13:1369–83.

[44] Keith JM, Adams P, Stephen S, Mattick JS. Delineating slowly and rapidly evolving DNA sequence segmentations. Bioinformatics 2007;23:657–63.

[45] Keith JM. Sequence segmentation. Methods Mol Biol 2008;452:207–19.

[46] Oldmeadow C, Keith JM. Model selection in Bayesian segmentation of multiple DNA alignments. Bioinformatics 2011;27:604–10.

[47] Green PJ. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 1995;82:711–32.

[48] Rameñosky VE, Maleser V, Roytberg MA, Tumanyan VG. DNA segmentation through the Bayesian approach. J Comput Biol 2006;7:215–31.

[49] Finklestein AV, Roytberg MA. Computation of biopolymers: a general approach to the analysis of DNA sequences. Comput Chem 2002;26:491–509.

[50] Futschik A, Hotz T, Munk A, Sieling H. Multiscale DNA partitioning: statistical evidence for segments. Bioinformatics 2008;24:2184–91.

[51] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via sequential importance sampling. Environ Model Assess 2009;14:577–84.

[52] Evans GE, Sofronov G, Keith J, Kroese D. Estimating change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[53] Sofronov G. Change-point modelling in biological sequences via the Bayesian adaptive independent sampler. J, International Conference on Telecommunication Technology and Applications; 2011. p. 22–126.

[54] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[55] Sofronov G. Change-point modelling in biological sequences via the Bayesian adaptive independent sampler. J, International Conference on Telecommunication Technology and Applications; 2011. p. 22–126.

[56] Oldmeadow C, Keith JM. Model selection in Bayesian segmentation of multiple DNA alignments. Bioinformatics 2011;27:604–10.

[57] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[58] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[59] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[60] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[61] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[62] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[63] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[64] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[65] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[66] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[67] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.

[68] Sofronov G, Evans G, Keith J, Kroese D. Identifying change-points in biological sequences via the cross-entropy method. Ann Oper Res 2011;189:155–65.