BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 9 2006, pages 1144–1146
doi:10.1093/bioinformatics/btl089

Genome analysis

BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data

J. C. Marioni[1,2,*], N. P. Thorne[1,2] and S. Tavaré[1,2]

[1]Hutchison-MRC Research Centre, Department of Oncology, Computational Biology Group, University of Cambridge, Hills Road, Cambridge and [2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge

Received on November 24, 2005; revised on February 7, 2006; accepted on March 7, 2006
Advance Access publication March 13, 2006
Associate Editor: Christos Ouzounis

ABSTRACT

Summary: We have developed a new method (BioHMM) for segmenting array comparative genomic hybridization data into states with the same underlying copy number. By utilizing a heterogeneous hidden Markov model, BioHMM incorporates relevant biological factors (e.g. the distance between adjacent clones) in the segmentation process.

Availability: BioHMM is available as part of the R library snapCGH which can be downloaded from http://www.bioconductor.org/packages/bioc/1.8/html/snapCGH.html

Contact: J.Marioni@damtp.cam.ac.uk

Supplementary information: Supplementary information is available at http://www.damtp.cam.ac.uk/user/jcm68/BioHMM.html

INTRODUCTION

Array Comparative Genomic Hybridization (aCGH) is an experimental technique used to detect DNA copy number changes across the genome. Test and reference samples are dyed with Cy3 and Cy5 respectively and hybridized to an array spotted with clones containing small regions of normal genomic DNA. The array is then scanned and image analysis software is used to measure the signal emitted from each clone. After preprocessing and normalization, the $\log_2$ ratio of the signal in the test and reference channels is determined for each clone. Subsequently, the aim is to segment these ratios into states, where all of the clones in a state have the same underlying copy number. Finally, a post-processing step is necessary to assign biological meaning to the different states.

From a statistical perspective, segmentation has received most attention and many different schemes have been proposed. These include calling clones as gained or lost if the ratio is above or below a set threshold, genetic local search algorithms (Jong *et al*., 2004), circular binary segmentation (Olshen *et al*., 2004), adaptive weights smoothing (Hupé *et al*., 2004) and hidden Markov models (HMMs) (Fridlyand *et al*., 2004). Two recent studies comparing the performance of segmentation schemes (Lai *et al*., 2005, Willenbrock and Fridlyand, 2005) have identified problems with existing methods. Despite the large number of methods, models have yet to be developed that take account of additional biological covariates such as the distance between adjacent clones or clone quality that are likely to affect the segmentation. In this article we describe a new segmentation scheme, BioHMM, which extends the HMM approach of Fridlyand *et al*. (2004) to take account of such information.

BioHMM is an integral part of the R library, snapCGH (Segmentation, Normalisation And Processing of aCGH data), which is a developer package obtainable from the BioConductor website (see Abstract for the address). This library also lets the user apply other segmentation schemes using common input and output data objects. In addition, snapCGH works seamlessly with limma objects (Smyth, 2005), enabling the use of pre-processing (and other) functions therein.

APPROACH

A hidden Markov model (HMM) is used to partition observations $y_1, y_2, \ldots, y_n$ made at 'locations' $t_1, t_2, \ldots, t_n$ into K (hidden) states, where $K << n$. In the context of aCGH data, the observations are (normalized) $\log_2$ ratios associated with clones arranged along a chromosome. For each clone $t_1, t_2, \ldots, t_n$ we know the start and end positions measured in base pairs and we assume that they are ordered along each chromosome by their start position. Subsequently, using the notation of Fridlyand *et al*. (2004) and Rabiner (1989), we can characterize a discrete (homogeneous) HMM where the observations are continuous as follows:

(1) The number of states, K, in the model. The states are denoted $S_1, \ldots, S_K$ and the chain is assumed to be irreducible; q_i denotes the actual state at position i $(1 \leq i \leq n)$.

(2) The initial state distribution, π, where $\pi_k = P(q_1 = S_k)$.

(3) The transition matrix, A, giving the probability of moving from one state to another, where

$$a_{lm} = P(q_{i+1} = S_m | q_i = S_l)$$

for $1 \leq i \leq n - 1$ and $1 \leq l, m \leq K$.

[*]To whom correspondence should be addressed.

(4) The distribution, b_k, of $\log_2$ ratios for clones in state S_k is assumed to be Gaussian with unknown mean and variance, i.e.

$$b_k \sim N(\mu_k, \sigma_k^2) \text{ for } 1 \leq k \leq K,$$

where we allow the possibility that the variance terms are equal.

Given the number of states, we can write the parameters to be estimated in the vector $\lambda = (A, B, \pi)$ where A, B and π denote the parameters in the transition matrix and the emission and initial distributions respectively. The optimal value of λ can be found by maximizing the likelihood, $L(\lambda|Y)$, where Y is a vector containing the observed $\log_2$ ratios. The likelihood is calculated by computing the forward–backward equations and the optimal values of the parameters are estimated using the EM algorithm, a Bayesian framework or a suitable numerical optimizer. Models with different numbers of states are (in general) distinguished using a criterion based upon the likelihood and the number of parameters, such as the AIC (Akaike, 1974). Finally, the Viterbi algorithm (Viterbi, 1967), or a similar method, is used to allocate clones to particular states.

We now describe a heterogeneous HMM, BioHMM, where the transition probabilities depend upon the distance, defined as the difference in midpoints, between adjacent clones. These distances are contained in a vector, x, which is of length $n-1$ since there are only $n-1$ transitions; information about the state to which the first clone is allocated is captured in the initial distribution. After constructing x, the information it contains can be incorporated into the transition probabilities of an HMM by considering a modified transition matrix, A_i, defined for $1 \leq i \leq n-1$ as

$$A_i = \begin{pmatrix} 1 - p_1(1 - e^{-f_i}) & p_1(1 - e^{-f_i}) \\ p_2(1 - e^{-f_i}) & 1 - p_2(1 - e^{-f_i}) \end{pmatrix},$$

where $f_i = x_i^r$ for $1 \leq i \leq n-1$ and $r \in \mathbb{R}$. We can rewrite this as

$$A_i = A_1 + e^{-f_i} A_2 \text{ for } 1 \leq i \leq n-1,$$

where

$$A_1 = \begin{pmatrix} 1 - p_1 & p_1 \\ p_2 & 1 - p_2 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} p_1 & -p_1 \\ -p_2 & p_2 \end{pmatrix}$$

We include the parameter r in the model because we expect there to be non-linear differences in the probabilistic structure of the transition matrix depending upon whether adjacent clones are very close or very far apart. We note that p_1 and p_2 have to be estimated subject to the constraint $0 \leq p_1, p_2 \leq 1$.

To estimate the value of these parameters (and the other parameters contained in λ) we used a numerical method to optimize the likelihood. To find initial parameter estimates, we used the methods described in Fridlyand *et al.* (2004) where applicable (for more information see Section 1.1.2 of the Supplementary Material); the initial estimate of r was set to be 1.

BioHMM is also capable of taking account of other biological information when segmenting the $\log_2$ ratios. This might include the length of each clone or a quality measure associated with each clone (perhaps obtained from the image analysis software). Some discussion of how we have incorporated additional covariates is given in Section 1.1.1 of the Supplementary Material.

We have only described the structure of the model when there are two underlying states. For details of the model when there are more states and an example of the application of BioHMM see the Supplementary Material.

DISCUSSION

In order to check that BioHMM yielded a sensible segmentation of real data, we applied it to the Coriell cell lines (Snijders *et al.*, 2001). This is a well-known cell line for which independently verified karyotype information is available. BioHMM correctly identified all of the changes which have been previously catalogued. For more information see Section 3.1 of the Supplementary Material, where we also provide an example of how transition probabilities vary depending upon the distance between the clones.

To further investigate the efficacy of BioHMM, and in particular to assess its performance relative to other segmentation schemes, a large amount of analysis will need to be carried out; this is beyond the scope of this application note.

One of the aims of both ourselves and Fridlyand *et al.* (2004) is to extend the HMM-based approach so that the whole genome can be segmented at once. Theoretically this is straightforward, but choosing suitable initial parameter estimates is difficult, and at present it is unclear how this should be done.

In conclusion, the approach adopted by BioHMM has a number of advantages over current segmentation schemes. Most importantly, by taking account of the distance between adjacent clones, BioHMM better models chromosomes where some regions are densely covered and others are covered at lower resolutions. In addition, BioHMM can be extended to take account of other biological covariates. We believe that these features, and the ease with which the model can be implemented, will ensure that BioHMM plays a useful role in the analysis of aCGH data.

ACKNOWLEDGEMENTS

The authors thank Jane Fridlyand, Nigel Carter and Heike Fiegler for useful discussions on segmenting aCGH data, Michael Smith for helpful suggestions on the programming of snapCGH, and Jessica Pole and Paul Edwards for allowing us to use unpublished data. JCM, NPT and ST are supported by Cancer Research UK. ST is a Royal Society/Wolfson Research Merit Award holder. Funding to pay the Open Access publication charges for this article was provided by CRUK.

Conflict of Interest: none declared.

REFERENCES

Akaike,H. (1974) A new look at statistical model identification. *IEEE Trans. Autom. Control*, 716–722.

Fridlyand,J. *et al*. (2004) Hidden Markov models approach to the analysis of array CGH data. *J. Multivar. Anal*., **90**, 132–153.

Hupé,P. *et al*. (2004) Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, **20**, 3413–3422.

Jong,K. *et al*. (2004) Breakpoint identification and smoothing of array comparative genomic hybridisation data. *Bioinformatics*, **20**, 3636–3637.

Lai,W.R. *et al*. (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, **21**, 3763–3770.

Olshen,A.B. *et al*. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

Smyth,G.K. (2005) Limma: linear models for microarray data. In Gentleman,R., Carey,V., Dudoit,S., Irizarry,R. and Huber,W. (eds), *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Springer, New York, pp. 397–420.

Snijders,A.M. *et al*. (2001) Assembly of microarrays for genome-wide measurement of DNA copy number. *Nat*. *Genet*., **29**, 4281–4286.

Viterbi,A.J. (1967) Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Trans*. *Inform*. *Theory*, **13**, 260–269.

Willenbrock,H. and Fridlyand,J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, **21**, 4084–4091.

Supplementary material for “BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data”

J.C. Marioni, N.P. Thorne and S. Tavaré

7th February 2006

1 Extending BioHMM

1.1 The structure of the model when there are more than two states

We now describe how BioHMM is used when there are more than two underlying states. Most of the components of the model can be extended in an obvious way using the framework described in the Approach section of the paper. Because of the constraints imposed upon the parameters in the transition matrix, its structure is slightly counter-intuitive. Hence we give a brief description of the structure of the transition matrix in the situation where there are three underlying states. The extension to four or more states is, indeed, straightforward.

Suppose that we wish to fit our heterogeneous HMM when there are three underlying states. We assume that there are n clones located on a chromosome and that a measure of the distance between adjacent clones is captured in a vector, x, of length $n-1$. We then define the transition matrix A_i for $1 \leq i \leq n-1$ as follows:

$$A_i = \begin{pmatrix} 1-(p_1+p_2)(1-e^{-f_i}) & p_1(1-e^{-f_i}) & p_2(1-e^{-f_i}) \\ p_3(1-e^{-f_i}) & 1-(p_3+p_4)(1-e^{-f_i}) & p_4(1-e^{-f_i}) \\ p_5(1-e^{-f_i}) & p_6(1-e^{-f_i}) & 1-(p_5+p_6)(1-e^{-f_i}) \end{pmatrix}$$

where $f_i = x_i^r$ and $r \in \mathbb{R}$. We can rewrite this as

$$A_i = A_1 + e^{-f_i} A_2 \text{ for } 1 \leq i \leq n-1$$

where

$$A_1 = \begin{pmatrix} 1-p_1-p_2 & p_1 & p_2 \\ p_3 & 1-p_3-p_4 & p_4 \\ p_5 & p_6 & 1-p_5-p_6 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} p_1+p_2 & -p_1 & -p_2 \\ -p_3 & p_3+p_4 & -p_4 \\ -p_5 & -p_6 & p_5+p_6 \end{pmatrix}$$

The parameters are estimated subject to the constraints $0 \leq p_k \leq 1$ (for $1 \leq k \leq 6$), $p_1 + p_2 \leq 1$, $p_3 + p_4 \leq 1$ and $p_5 + p_6 \leq 1$.

1.1.1 Adding additional covariates and the selection of f_i

At present, we have only examined the effect of incorporating a distance covariate in the model. However, BioHMM is capable of taking account of additional biological covariates such as the length of each clone or the quality of the $\log_2$ ratio associated with each clone (perhaps obtained from the image analysis software). We now give a brief overview of how such information might be included in the model.

Given such information, we can define a covariate matrix, X, of dimension $(n-1) \times c$ where c is the number of covariates. We impose the constraint on X that its first column must contain a measure of the distance between clones. In order to incorporate these covariates into the HMM we can use a modified version of A_i.

There are many choices for the structure of f_i depending upon how we think the covariates are related. Below, we describe two of the simpler options:

1. If we assume that the covariates act independently of one another we might set f_i to be an additive function (with no interaction terms). For example, $f_i = \sum_{j=1}^{c} x_{ij}$.
2. Alternatively, if we assume that there is an interaction between different covariates, we might set f_i to be a multiplicative function. For example, $f_i = \prod_{j=1}^{c} x_{ij}$.

In the current implementation of BioHMM we use a modified version of the second option since we expect there to be an interaction between different covariates (at least between those that have been considered to date - see the discussion below on incorporating spot quality weight as a covariate). The modified structure we used is as follows:

$$f_i = \prod_{j=1}^{c} x_{ij}^{r_j}$$

where $r_j \in \mathbb{R}$ for $j = 1$ and $r_j = 1$ for $j > 1$.

We use the parameter r_1 when modelling the "distance" covariate (recall that the first column in X must contain a measure of the distance between adjacent clones) because we expect there to be non-linear differences in the probabilistic structure of the transition matrix depending upon whether adjacent clones are close or far apart. For other covariates we do not include such parameters as we expect the effect of these covariates on the transition probabilities to be linear.

As an example, we consider how spot quality might be incorporated into BioHMM. We assume that spot quality is measured on a continuous scale between 0 and 1 (where the lower the score the poorer the quality of the spot), and we use this value as the j^{th} of c covariates. Therefore, if the $(i+1)^{th}$ clone along a chromosome is of poor quality, f_i will be near to 0 irrespective of the values associated with any of the other $c-1$ covariates. As f_i tends to 0, e^{-f_i} will tend to 1 and so A_i will tend towards the identity matrix. In other words, because we do not have much confidence in the measurement associated with the $(i+1)^{th}$ clone, BioHMM decreases the probability that this clone is in a different state regardless of the distance to its neighbours.

1.1.2 Parameter estimation

We estimate the parameters using the Nelder-Mead numerical optimisation algorithm [2] which is written in C and called from within *snapCGH*. This method determines a local optimum and the corresponding parameter values, and is known to perform well for non-differentiable functions.

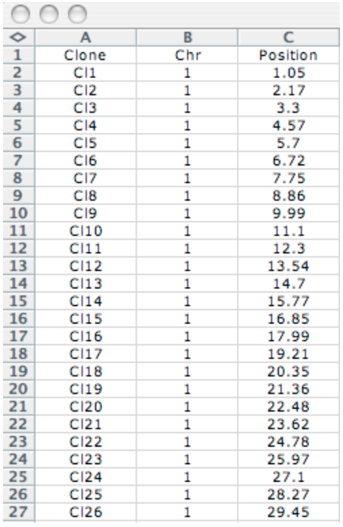
In general, we use the initial parameter estimates described in [1]. For the emission distribution, B, the estimates of the means and variances for a given number of states are taken to be the mean and variance of the states determined by applying a Partitioning Around Medoids (PAM) algorithm to the data and for a given K, the initial estimates for π are all set as $1/K$. Additionally, the initial estimate of the rate parameter r_1 is set to be 1. However, for the parameters in A, the transition matrix, the initial diagonal estimates are all 0.95 (rather than 0.99 in [1]) and given the number of parameters, K, the off-diagonal terms are $0.05/K$ for $K > 1$.

2 Using BioHMM

BioHMM is an integral part of the *snapCGH* library. In this section we provide an overview of how *snapCGH* might be used and include examples of the format of the clone order and covariate information required. In section 3 we give the commands that were used to apply BioHMM to a real dataset.

2.1 Reading in and preprocessing the data

snapCGH uses functions from the *limma* library to read in the data and create a typical *limma* type object. The next step is to add information about the genomic location of clones to this object. In order to do this it is necessary to create a clone order information file. This should be a dataframe with rows corresponding to individual clones arranged by their starting position along the genome. There should be three columns: the first column should be called "Clone" giving the name of the clone, the second column should be called "Chr" giving the chromosome on which a clone is located and the third column should be called "Position" giving the midpoint of a clone (in megabases).



◇	A	B	C
1	Clone	Chr	Position
2	Cl1	1	1.05
3	Cl2	1	2.17
4	Cl3	1	3.3
5	Cl4	1	4.57
6	Cl5	1	5.7
7	Cl6	1	6.72
8	Cl7	1	7.75
9	Cl8	1	8.86
10	Cl9	1	9.99
11	Cl10	1	11.1
12	Cl11	1	12.3
13	Cl12	1	13.54
14	Cl13	1	14.7
15	Cl14	1	15.77
16	Cl15	1	16.85
17	Cl16	1	17.99
18	Cl17	1	19.21
19	Cl18	1	20.35
20	Cl19	1	21.36
21	Cl20	1	22.48
22	Cl21	1	23.62
23	Cl22	1	24.78
24	Cl23	1	25.97
25	Cl24	1	27.1
26	Cl25	1	28.27
27	Cl26	1	29.45

Figure 1: An example of a clone information input file for BioHMM created using Excel. The file is saved in "txt" format.

After this information has been added, the data can be background corrected and normalised using standard functions within *limma*. Subsequently, before the data can be segmented it is necessary to tidy them up in order to account for replicated clones and for clones which are not mapped to the human genome. All of these operations can be carried out within *snapCGH*.

2.2 The format of the covariate information

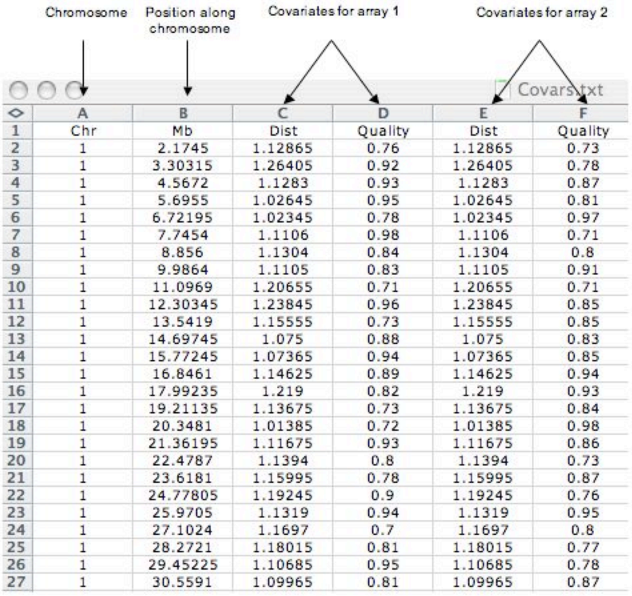
The covariate information is read in as a dataframe with $Ac + 2$ columns and $N - t$ rows, where c is the number of covariates, A is the number of different arrays (or samples), N is the total number of clones located along the genome and t is the number of chromosomes on which clones are located. The first two columns in the dataframe should give the chromosome on which a clone is located ("Chr") and the position of a clone along a particular chromosome ("Mb"). The clones should be ordered according to their position along the genome - however, information about the clone located nearest the left hand telomere should be excluded for each chromosome. This is necessary because for each chromosome with n clones located on it there are only $n-1$ transitions.

For array 1 the covariate information is then contained in the next c columns in the dataframe (columns 3 to $3+(c-1)$). Similarly, for array 2 the covariate information is contained in columns $3+c$ to $3+(2c-1)$ and so on for the remaining arrays. We note that columns 3, $3+c$, $3+2c\ldots$ must contain a measure of the distance between adjacent clones (in megabases). If no information about the distance between clones is available (equivalent to the situation where all clones are equally spaced) we can model this in BioHMM by setting the distance between all of the clones to be one. The covariate information can be created using Excel where it should be saved as a "txt" file before being read into R.

As an example of such a file, suppose we want to include the following covariates in our model:

1. The distance between the midpoint of adjacent clones measured in megabases;
2. A score of clone quality obtained from the image analysis software. This is assumed to be a value between 0 and 1, where a score close to 0 indicates that the spot is of poor quality, while a score close to 1 indicates that the spot is of high quality.

The corresponding "txt" file should then have a similar structure to the extract shown below.



◇	A	B	C	D	E	F
1	Chr	Mb	Dist	Quality	Dist	Quality
2	1	2.1745	1.12865	0.76	1.12865	0.73
3	1	3.30315	1.26405	0.92	1.26405	0.78
4	1	4.5672	1.1283	0.93	1.1283	0.87
5	1	5.6955	1.02645	0.95	1.02645	0.81
6	1	6.72195	1.02345	0.78	1.02345	0.97
7	1	7.7454	1.1106	0.98	1.1106	0.71
8	1	8.856	1.1304	0.84	1.1304	0.8
9	1	9.9864	1.1105	0.83	1.1105	0.91
10	1	11.0969	1.20655	0.71	1.20655	0.71
11	1	12.30345	1.23845	0.96	1.23845	0.85
12	1	13.5419	1.15555	0.73	1.15555	0.85
13	1	14.69745	1.075	0.88	1.075	0.83
14	1	15.77245	1.07365	0.94	1.07365	0.85
15	1	16.8461	1.14625	0.89	1.14625	0.94
16	1	17.99235	1.219	0.82	1.219	0.93
17	1	19.21135	1.13675	0.73	1.13675	0.84
18	1	20.3481	1.01385	0.72	1.01385	0.98
19	1	21.36195	1.11675	0.93	1.11675	0.86
20	1	22.4787	1.1394	0.8	1.1394	0.73
21	1	23.6181	1.15995	0.78	1.15995	0.87
22	1	24.77805	1.19245	0.9	1.19245	0.76
23	1	25.9705	1.1319	0.94	1.1319	0.95
24	1	27.1024	1.1697	0.7	1.1697	0.8
25	1	28.2721	1.18015	0.81	1.18015	0.77
26	1	29.45225	1.10685	0.95	1.10685	0.78
27	1	30.5591	1.09965	0.81	1.09965	0.87

Figure 2: An example of a covariate input file for BioHMM created using Excel. This file is saved as a "txt" file.

After the covariate information has been read in we can apply BioHMM (and the other segmentation methods available within *snapCGH*) to the data.

3 Examples of the application of BioHMM

We now illustrate the output obtained when BioHMM is applied to the Coriell cell lines [4], an (unpublished) breast cancer dataset containing 10 samples and a dataset simulated using a novel scheme that is included within *snapCGH*. The sole covariate used in each example is the distance between clones. For the breast cancer dataset we contrast the segmentation achieved using BioHMM with those obtained using other schemes. We do not claim that this is a thorough comparison of BioHMM with other segmentation methods, but the example described illustrates a situation where we expect our model to yield a different segmentation to current methods.

3.1 Real data

The Coriell cell lines

As described in the Discussion section of the paper, the Coriell cell lines [4] are commonly used in assessing the efficacy of methods for segmenting aCGH data since independently verified karyotype information is available for all major transitions. In common with the homogeneous HMM [1] BioHMM was able to identify all such changes as well as finding a number of other transitions whose veracity has not been established. Examples of some of the changes correctly identified by BioHMM are shown in Figure 3.

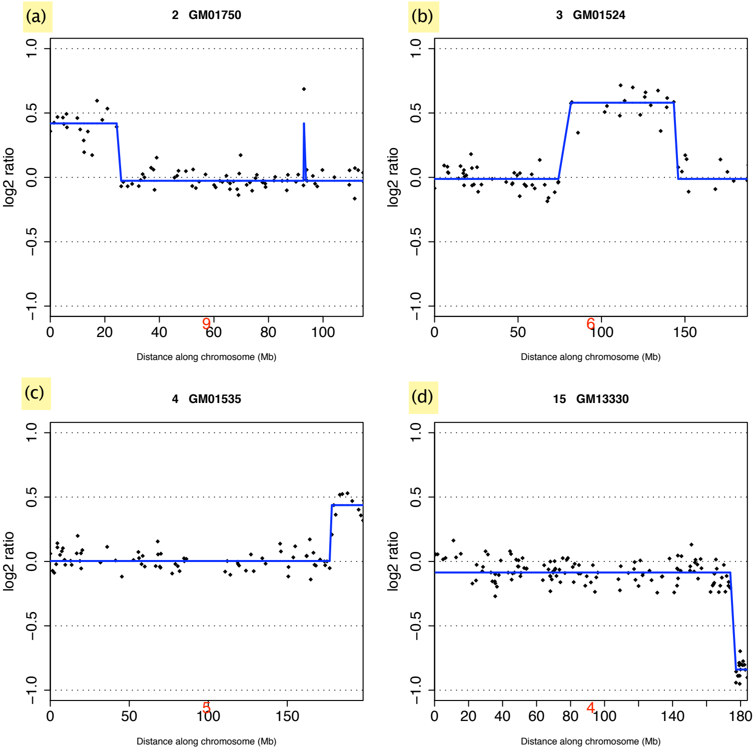


Figure 3: Examples of the application of BioHMM to the Coriell cell lines. Fig. (a) shows a single clone aberration on the p-arm. It is not possible to verify whether this is a biologically meaningful aberration or simply a mismapped clone or experimental noise.

The Coriell cell line data only has a small amount of noise and the transitions are very well defined - in this sense it is atypical of much array CGH data. We now give an example of how BioHMM can be applied to a much noisier (and therefore more difficult to accurately segment) breast cancer dataset. We also give the R commands that were used to fit BioHMM to this dataset.

The breast cancer dataset

The commands used to apply BioHMM to this dataset are given below. For more details on how the functions used below operate see the relevant R help files.

```
library(snapCGH)
targets <- readTargets("targets.txt")
RG <- read.maimages(targets$FileName, source = "genepix")

RG <- read.clonesinfo("cloneinfo.txt", RG)
# This adds the information about the location of each clone on the genome to the
  RG object

RG$printer <- getLayout(RG$genes)
RG1 <- backgroundCorrect(RG, method = "minimum")
MA1 <- normalizeWithinArrays(RG1, method = "median")

MA1$design <- rep(-1,10)
# A -1 in the design matrix indicates that the test sample is in the green channel

MA2 <- processCGH(MA1, method.of.averaging="mean")
# This averages replicated clones and removes unmapped clones as well as tidying up
  the dataset

covariates <- read.table("covariates.txt", header = T, sep = "\t")
covariates <- as.data.frame(covariates)
# These commands read in the covariate information into the R session

BioHMM <- fitBioHMM(MA2, covariates = covariates, numiter = 30000)
# This command runs BioHMM

BioHMM.merged <- mergeStates(BioHMM, MergeType = 2, minDiff = 0.2)
# For merging states whose means are too close to each other

plotSegmentedGenome(BioHMM.merged, array = 1, chrom.to.plot = 8)
# A function for plotting the resulting output
```

We illustrate the (unmerged) segmented output (for a particular chromosome and sample) obtained using DNAcopy [3], a homogeneous HMM and BioHMM. The coverage of the clones is 1Mb for the majority of this chromosome, but for a small region on the short arm the clones are tiled (i.e. they overlap) and so the resolution is much higher. This is the sort of clone layout where we expect the distance between clones to impact upon the segmentation.

In Figure 4 we illustrate the segmented output for the whole of the chromosome. We can observe that there are differences in the segmentations obtained which nicely illustrate a number of features associated with each scheme. We can observe that in each of the HMM-based segmentation schemes, the first clone is allocated to a different state than the second clone. This is not the case in the

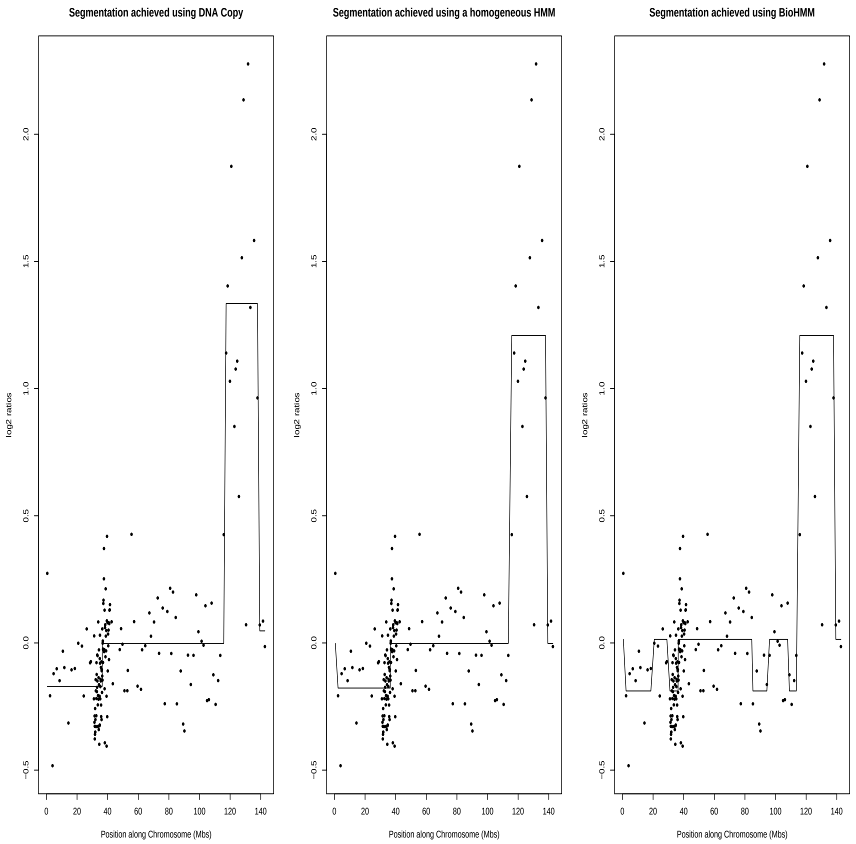


Figure 4: Data segmented using different schemes. The segmentations obtained using DNAcopy, a homogeneous HMM and BioHMM (from left to right respectively) are superimposed upon the actual data. The segments and the actual data are plotted against the physical location of the clones on the chromosome.

segmentation suggested by DNAcopy because, by its construction, it is not able to allocate a single clone to a different state than its neighbours.

Other than this, we can observe that the segmentations suggested by DNAcopy and the homogeneous HMM are extremely similar with only minor differences in the locations of the breakpoints. However, the segmentation obtained using BioHMM is clearly different. In particular, we can observe four transitions which are not seen in the other segmentations. While it is difficult to infer why a difference exists between the segmentations obtained using BioHMM and DNAcopy, the reasons for the difference in the segmentation between BioHMM and the homogeneous HMM are explained by differences in the transition matrices.

The transition matrix for the homogeneous HMM, A^{hom}, is as follows:

$$A^{\text{hom}} = \begin{pmatrix} 0.973 & 0.015 & 0.012 \\ 0.020 & 0.980 & 0.000 \\ 0.065 & 0.000 & 0.935 \end{pmatrix}$$

The transition matrix obtained when BioHMM is applied, A^{Bio}, is:

$$A_i^{\text{Bio}} = \begin{pmatrix} 0.782 & 0.218 & 0.000 \\ 0.323 & 0.627 & 0.050 \\ 0.081 & 0.000 & 0.919 \end{pmatrix} + e^{-x_i^{1.46}} \begin{pmatrix} 0.218 & -0.218 & -0.000 \\ -0.323 & 0.373 & -0.050 \\ -0.081 & -0.000 & 0.081 \end{pmatrix}$$

for $1 \leq i \leq n-1$.

The first thing to note is that $r = 1.46$ - this is a relatively large value which suggests there is quite a degree of heterogeneity in the model fitted using BioHMM. We can observe this more clearly if we compare a typical transition matrix in a region where the coverage is 1Mb, with a transition matrix in the tiled region.

$$A^{\text{1Mb}} = \begin{pmatrix} 0.790 & 0.209 & 0.000 \\ 0.311 & 0.641 & 0.048 \\ 0.078 & 0.000 & 0.921 \end{pmatrix} \text{ and } A^{\text{tiled}} = \begin{pmatrix} 0.999 & 0.000 & 0.000 \\ 0.013 & 0.985 & 0.002 \\ 0.003 & 0.000 & 0.997 \end{pmatrix}$$

We also note that by using the likelihood ratio test ($p = 0.013$) we can conclude that the model fitted using BioHMM is statistically superior to the homogeneous HMM.

Clearly, the probability of a transition out of the current state is much greater in regions where the clones are located further apart than in tiled regions and correspondingly, this explains the different segmentations obtained using a homogeneous HMM and BioHMM.

3.2 The simulated dataset

We illustrate the segmentation for a single chromosome and we focus in on a region which illustrates the effect we expect BioHMM to have.

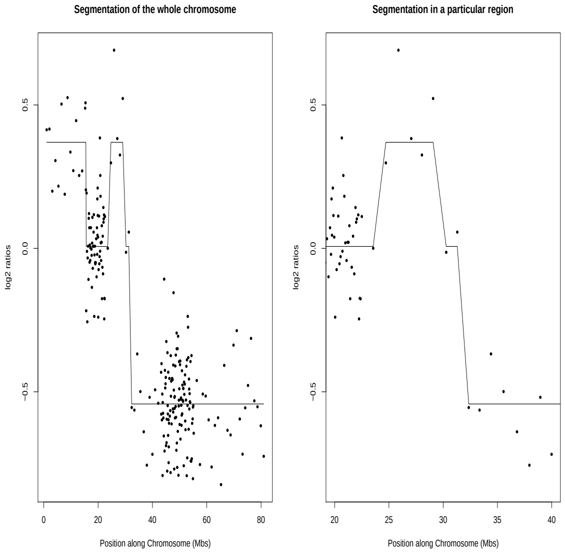


Figure 5: Simulated data segmented using BioHMM for the whole chromosome (left) and for a particular region of the chromosome (right). The segments and the data are plotted against the physical location of the clones on the simulated chromosome.

This example shows how BioHMM sensibly takes account of the distance between clones when carrying out a segmentation. In the right hand plot we can observe that in the tiled region (between 20 and 23 Mbs) there is quite a lot of variation in the $\log_2$ ratios. However, because the clones are very close together, the probability of a transition between states for these clones is small and hence all of these clones are allocated to the same state. This contrasts with the clones in the untiled region (from 23Mb onwards) where there is a much lower probability of remaining in the current state and thus there are many more transitions.

References

[1] Fridlyand, J., Snijders, A.M., Pinkel, D., Albertson, D.G., Jain, A.N. (2004) Hidden Markov models approach to the analysis of array CGH data, *Journal of Multivariate Analysis*, **90**, 132-153.

[2] Nelder, J.A., Mead, R. (1965) A simplex algorithm for function minimization, *Computer Journal*, **7**, 308-313.

[3] Olshen, A.B., Venkatraman, E.S., Lucito, R., Wigler, M. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data, *Biostatistics*, **5**, 557-572.

[4] Snijders, A.M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, N., Hamilton, G., Hindle, A.K., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J.P., Gray, J.W., Jain, A.N., Pinkel, D., Albertson, D.G. (2001) Assembly of microarrays for genome-wide measurement of DNA copy number, *Nature Genetics*, **29**, 4281-4286.