Thermodynamically ultra-fastened DNA regions mapping based on hidden Markov modeling

Colin Veal*† Alexandros Giavaras ‡Anthony Brookes§

Abstract

A mapping of thermodynamically ultra-fastened (TUF) DNA regions based on a hidden Markov model (HMM) is discussed. Initial results indicate that the developed HMM is capable of capturing the TUF regions formed in the DNA strand. This is verified by visually assessing the Viterbi paths generated by the model.

Keywords: Thermodynamically ultra-fastened, DNA, hidden Markov model,

1 Introduction

Thermodynamically ultra-fastened (TUF) regions are stretches of the DNA which fail to denature even after the application of extreme melting conditions [12]. This behavior effectively reduces the amplification efficiency in these regions. It has been reported that TUF regions contain a core sequence which exhibits an increased GC concentration relative to the surrounding DNA. It is in fact these locally concentrated spikes of GC content which is believed to remain duplexed despite the application of denaturation processes [12].

Computational modelling and analysis of TUF regions requires the ability to somehow identify these regions in the DNA strand. However, to the best of our knowledge, such tools do not yet exist. Visually, identifying and labeling TUF regions, although feasible, is time consuming and error prone to say the least.

The aim of this work [1] is twofold; the development of a mathematical approach so that a mapping of TUF regions can be created and the analysis of the TUF core regions. Such a tool will allow the labeling of these regions so that these can identified and further analysed. In this regard, we employ a hidden Markov modelling methodology. Concretely, our approach uses two sequences; a sequence that it underwent WGA treatment (sample

*Department of Genetics and Genome Biology, University of Leicester, UK. Email: cdv1@leicester.ac.uk.
†Corresponding author.
‡Department of Genetics and Genome Biology, University of Leicester, UK.
§Department of Genetics and Genome Biology, University of Leicester, UK.

[1]The software developed under the present study can be found at https://github.com/pockerman/hmmtuf_app

m605), and one that was not treated (sample m585). This is necessary in order to amplify the existence of TUF regions. In whole genome amplified (WGA) read-depth sequencing samples, TUF regions are often represented as regions of low coverage, resembling deletions. This resemblance actually makes it very difficult to computationally distinguish with copy deletions. Thus, the developed HMM classifies segregated chromosome regions into states depending on their average read-depth count. Although the exact characteristics in terms of read-depth of such a state are not known, the assumption is that a low read-depth observed in a WGA sample in combination of a normal one observed in the same region for a non-WGA treated sample is indicative of TUF. Using this read-depth based characterization, we can distinguish commonly found behavior in sequences such single copy or full copy deletion. In terms of the HMM, the TUF regions can simply be modelled as an extra state. Once a characterization of the DNA regions is available in terms of TUF, we can further process these regions in order to understand their structure.

The remaining of this work is organised as follows. Section 2 briefly describes the general principles underlying hidden Markov models. Concretely, we focus on discrete and time invariant models. Section 3 discusses the particularities of our approach towards establishing a hidden Markov model for TUF regions. Section 4 discusses our efforts to understand the underlying nucleotide structure of TUF core regions. Section 5 presents some initial mappings extracted using the developed HMM. Finally, section 6, presents a discussion on the developed methodology and indicates possible future work.

2 Hidden Markov model

A hidden Markov model (HMM) is a probabilistic framework that uses two interrelated probabilistic mechanisms; a Markov chain of a finite number of states, N, and a set of random functions each associated with a respective state [11]. The set of discrete states is denoted by $S = \{S_0, S_1, \cdots, S_{N-1}\}$. At a given time instant, the system is assumed to be in some state and an observation is generated by the random function corresponding to this state [11]. State transitioning occurs according to a transition probability matrix $\mathbf{A}$. Within the HMM framework, an observer only sees the random output generated by the random functions corresponding to the states and not the states themselves. Thus, the state at which the system is in can only be probabilistically inferred.

We use more or less standard notation and denote with $q_n \in S$ the state of the system under consideration at the discrete time instance n. A sequence of states, each of which belongs in S, is denoted with Q; $Q = \{q_1 q_2, \cdots q_T\}$. Moreover, a sequence of observations is denoted with O; $O = \{o_1 o_2, \cdots o_T\}$. Overall, an HMM is characterized by the following parameters, see [8] and [11]

- $\mathbf{A}$ a probability transition matrix
- $\mathbf{B}$ a probability emission matrix

- $\boldsymbol{\pi}$ an initialization vector

Each a_{ij} of $\mathbf{A}$ expresses the probability of transitioning to state j given that the previous state was i namely

$$a_{ij} = P(q_n = j | q_{n-1} = i), \forall i, j \in S \tag{1}$$

Equation 1 expresses the assumption that the system states form a Markov chain [8]. In other words, the current system state depends only on the previous state. Since the a_{ij}s represent probabilities, the following conditions should be respected [11]

$$a_{ij} \geq 0, \sum_j a_{i,j} = 1 \tag{2}$$

Similarly, each element b_{jk} of the emission matrix $\mathbf{B}$ specifies the probability that at time instant n and state j, the observation is o_k [11]:

$$b_{jk} = P(O_n = o_k | q_n = j) \tag{3}$$

We have the following constraints for the $\mathbf{B}$ matrix

$$b_{jk} \geq 0, \sum_k b_{jk} = 1 \tag{4}$$

An HMM does not require the number of states is the same as the number of observation symbols. Finally, the vector $\boldsymbol{\pi}$ provides the probability distributions at time $n = 0$ meaning

$$\pi_j(0) = P(q_0 = j) \tag{5}$$

Collectively, we denote an HMM using the letter λ:

$$\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) \tag{6}$$

Finally, we assume that we are dealing with a time invariant system. In other words, the transition probability matrix remains constant [8] and [11].

Hidden Markov models have been used quite extensively in bioinformatics; copy number variation detection [3], [13] and [2], analysis of of of array CGH data [5], analysis of profile series [10]. A general review of hidden Markov modelling in relation to bioinformatics is given, for example, in [11]. In this work, we develop a hidden Markov model in order to establish a mapping for thermodynamically ultra-fastened DNA regions. Out methodology is described in the next section.

3 Hidden Markov model for TUF regions

In this section we describe a hidden Markov model for mapping TUF regions. We develop the model by using the pomegranate [2] Python library. Our approach uses two sequences; a sequence that it underwent WGA treatment (sample m605), and one that was not treated (sample m585). This is necessary in order to amplify the existence of TUF regions. Figure 1 shows a snapshot of the amplification in the WGA sample compared to the non-treated one as these are viewed in the IGV browser [9].

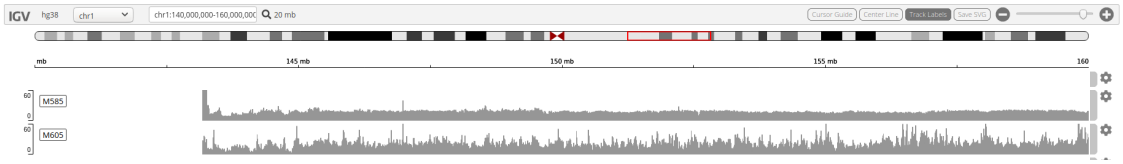

FIGURE 1: m605 and m585 samples.

The developed model assumes the following set of discrete states:

- Deletion
- TUF
- Normal copy (two states see below)
- Duplication
- TufDup
- Gap

The Gap state corresponds to the case where there is no base present in either of the sequences used. The TUF state assumes a low WGA sample mean when compared to normal sample mean for the non-WGA sample. However, this does not fully capture the spectrum of the data, see figure 8. Thus, we introduce the TufDup state in order to represent data where the WGA sample mean is rather small whilst the non-WGA sample mean is large enough to assume that this is a Duplication state.

As mentioned in the previous section, a hidden Markov model assumes that the system in hand can be in a state from a specified set S. This set of states can be assumed a priori implying some knowledge of the data. Examples of this methodology are given in [3] and [13] where six states are used. Another approach is to use a clustering technique in order to determine the optimal number e.g. [5] and [7]. In the latter approach, each cluster is assumed to represent a state. Clusters being very similar under some metric can be merged together. A recent short survey on clustering techniques can be found in [6].

[2] https://github.com/jmschrei/pomegranate

One advantage of the clustering approach is that it allows for an educated guess about the optimal number of states represented in the data. Furthermore, it allows for an estimation parameters of the distributions that the HMM framework requires rather than resorting to heuristic assumptions. However, clustering can be as good as the data allows to distinguish the various states assumed. For example, TUF and single copy deletion are states that, at least with the data used in this work, are difficult to differentiate. Frequently used states in CNV studies are full and single copy deletion, normal and duplication. In this work, we also assume the existence of TUF and Gap states.

After determining the states that will be used in the HMM, we need to establish the appropriate probability distribution that best models each state in terms of emission probabilities. This, in general, seems to be more important than how one is modeling the transition probability matrix **A**, [8]. There is a variety of methods to achieve this. The simplest is to assume a priori a given probability mass function with given parameters. Another approach is to use an estimation technique such as histograms, kernel estimation or clustering.

In this work, we assume that the states follow a two dimensional Gaussian distribution. The exception to this is the Gap state where we use a uniform distribution. The empirical distributions that we compute when investigating the data, this is the mean read-depth count per window, suggest that this assumption is not unreasonable. Figures 2, 3 and 4 plot the empirical distributions of the mean read-depth count per window of the data that was used to establish the distributional parameters for full copy deletion, single copy deletion and duplication states.

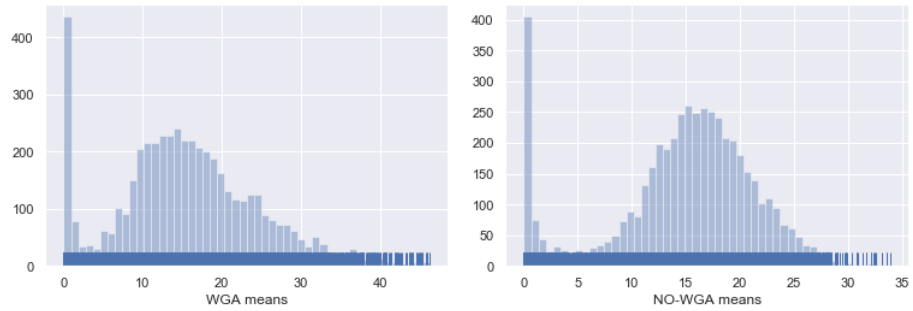

FIGURE 2: Full copy deletion histogram for WGA and non-WGA samples.

We estimate the parameters for these Gaussian distributions as follows. We cluster by using using a Gaussian mixture model (GMM) [3] [4] a data set which contains manually identified portions of the DNA that match the assumed states . The data set corresponds to small regions from chromosome 1 that contain the states that the model assumes. These regions are then discretised into non-overlapping windows each of which has size 100 bases. We calculate the read-depth count means for the two samples, i.e. WGA and non-WGA, and then apply a cutoff filter to exclude outliers. The filter is simply a threshold on the

[3]We use the scikit-learn implementation. See https://scikit-learn.org/stable/ for more details.

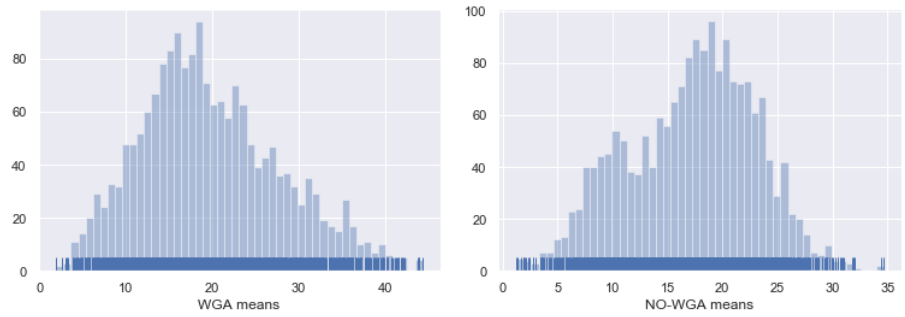

FIGURE 3: Single copy deletion histogram for WGA and non-WGA samples.

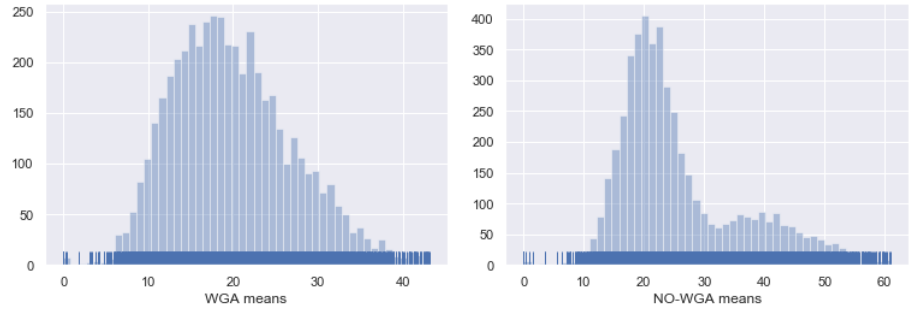

FIGURE 4: Duplication histogram for WGA and non-WGA samples.

means. Hence, a window is assumed as an outlier if either $\mu_{WGA} > 140$ or $\mu_{NWGA} > 120$. Where μ_{WGA} is the mean for the WGA sample and μ_{NWGA} is the mean for the non-WGA sample. The ensuing windows, form the input for the GMM . We also investigated more traditional approaches like K-Means and PAM. However, these techniques tend to create equally sized clusters. This is something that we do not anticipate to be the case (for example the Normal state is expected, in general, to dominate the data). A GMM approach allows for more flexibility on the shapes of the clusters whilst we can use the parameters of the ensued Gaussian distributions, i.e. μ_{WGA}, μ_{NWGA} and $\mathbf{\Sigma} = diag(\sigma^2_{WGA}, \sigma^2_{NWGA})$ where σ_i is the window standard deviation for the WGA and non-WGA sample, into the HMM model. In GMM clustering the hard cluster assignment of K-means, is changed into a soft one [4]. Note that the windows which have been identified to contain gaps are excluded from the clustering calculations however they are kept in the final model. Figure 5 shows the clustered data when using five clusters. Only four clusters are actually visible. The cluster that represented deletion was dropped in favor of the red cluster in the figure. Moreover, the single copy deletion cluster was also dropped. The yellow cluster is used to extract the parameters for the Duplication state whilst the pink and blue are used to model two different Normal states labelled as Normal-I and Normal-II. The TUF state is represented as a mixture model with two components each of which is represented as a two dimensional Gaussian distribution. Each component is weighted using a coefficient of $1/2$. Figure 6 shows the clustering for identifying the properties of the TUF state. The green component

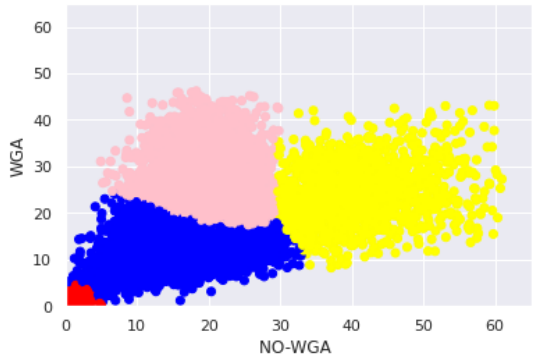

FIGURE 5: GMM clustering with five clusters.

shown in figure 6 is used in order to initialize the TUF state.

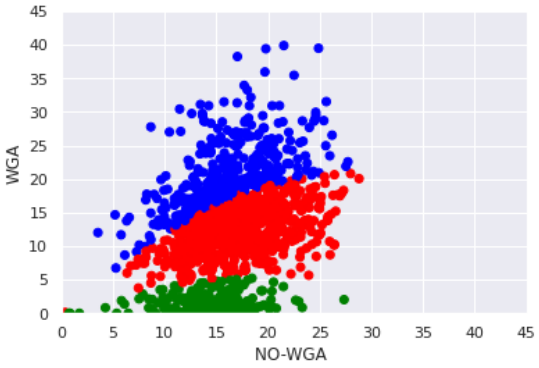

FIGURE 6: GMM clustering for TUF state with three clusters.

In order to model the Gap state we assume that both components follow a uniform distribution $U(-999.5, -998.5)$. Our intention is to make this state to stand out from the rest so that the model is forced to select this when a gap window is found (see section 5).

The TufGap state is introduced after calibrating the model on chromosome 1. This is necessary as it is difficult to identify representative data for every state. Calibration is done by applying the model on various regions and extracting the Viterbi path. The resulting path is then visually evaluated by loading both the region samples and the path on the IGV browser [9]. Figure 7 shows the predicted states for chromosome 1 and region $[1-20]\times 10^6$ using the non-calibrated HMM model. The red spikes correspond to TUF windows. Figure 8 shows the classification of the windows [4] achieved by the calibrated model on region $[1-20]\times 10^6$. The non-calibrated model did not include the TufDup state. This caused the purple dots to be classified as duplication (shown in green).

The TufDup state is also modeled using a two-dimensional Gaussian distribution. The parameters for the distribution are evaluated based on figure 8 as follows. The WGA

[4]The windows are represented by the WGA and NWGA means

,

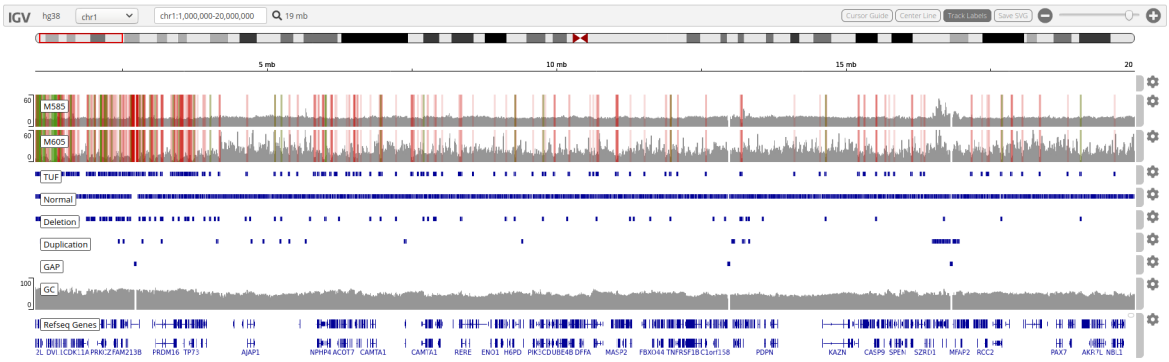

FIGURE 7: Viterbi path classification of windows.

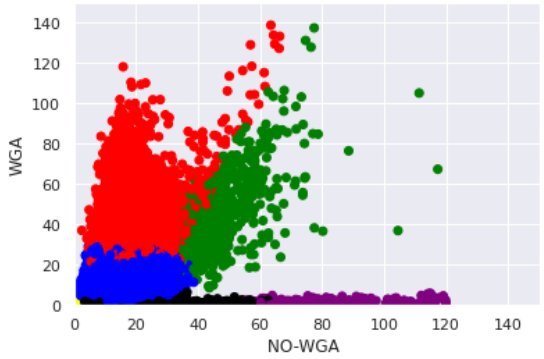

FIGURE 8: Viterbi path classification of windows for calibrated model.

component of the two dimensional distribution is using the mean and variance from the neighbouring TUF state i.e. black dots in figure 8. The NWGA component uses $\mu_{NWGA} = 85$ which roughly corresponds to the middle of the purple dots in figure 8. The variance σ^2_{NWGA} is set according to the formula 7

$$\sigma^2_{NWGA} = \sigma^2_{NWGA,TUF} + 0.3\sigma^2_{NWGAG} \tag{7}$$

where $\sigma^2_{NWGA,TUF}$ is the variance of the non-WGA sample from the nearby TUF region and σ^2_{NWGAG} is the variance of the non-WGA sample from the Duplication cluster i.e. green dots in figure 8. Thus, we assign portion of the σ_{NWGAG} to the variance of the non-WGA sample. The factor 0.3 was determined by plotting the contours of the two ensued distributions for TUF and TufDup and checked whether their contours mixed. We chose the parameter such that the two distributions barely mix with each other.

The HMM also requires as input the initialization vector $\boldsymbol{\pi}$ and the transition matrix $\mathbf{A}$, see equation 6. For the former we assume a uniform probability for every state i.e. every state is equally likely to initiate the sequence of hidden states. Hence,

$$\pi_i = \frac{1}{|S|}, \quad \forall i \in S \tag{8}$$

where $|S|$ denotes the number of discrete states. For the latter, we assume that every state can transition to any other state including itself. However, we assign a significantly higher probability to the latter scenario than the former. In other words, we assume that the model is more likely to stay in a given state than transitioning to another. This is summarized in equation 9

$$\mathbf{A} = \begin{bmatrix} 0.85 & 0.025 & 0.025 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.85 & 0.025 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.85 & 0.85 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.85 & 0.025 & 0.85 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.85 & 0.025 & 0.025 & 0.85 & 0.025 & 0.025 \\ 0.025 & 0.85 & 0.025 & 0.025 & 0.025 & 0.85 & 0.025 \\ 0.025 & 0.85 & 0.025 & 0.025 & 0.025 & 0.025 & 0.85 \end{bmatrix} \tag{9}$$

In summary, the used hidden Markov model is as follows

- $\pi_i = \frac{1}{|S|}, \quad \forall i \in S$
- Gap state $G \sim U(-999.5, -998.5)$
- TUF and TufDup states $\sim \sum_{i=1}^{2} c_i N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), c_i = 1/2$
- Every other state $S \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Figure 9 shows the HMM model with the transition probabilities used in a graphical form. We remark however, that the framework we use is flexible enough to assume different parametric models and add or remove states.

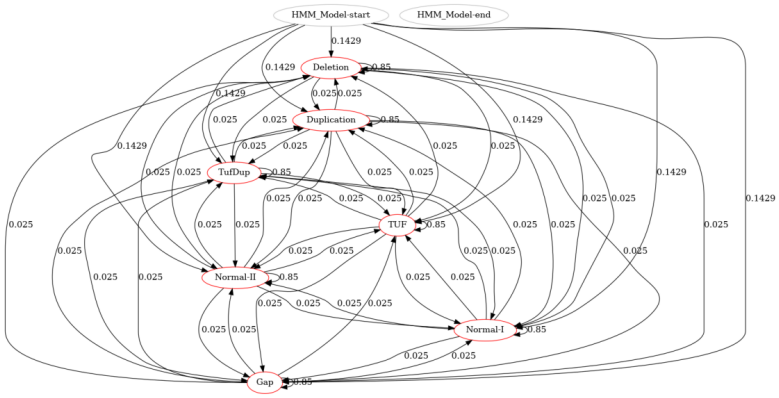

FIGURE 9: States and transition probabilities for HMM model.

4 Towards TUF core analysis

Once a characterization of the DNA sequence in terms of TUF regions is available, one can further question what is the underlying nucleotide structure of these TUF regions. In order to do so, we process the Viterbi paths using *SPADE* [5].

This is a software for exploring periodic repeat regions from large genomic and protein data resources. *SPADE* first extracts multiple sequence entries from an input file (GenBank or FASTA format) and identifies sequence type (DNA or protein) for each entry. Each sequence entry is scanned by a sliding window to count k-mers and highly repetitive regions are extracted. The sequence periodicity of each highly repetitive region is then evaluated based on position-period matrix that cumulatively plots distance between neighboring same k-mers and their sequence positions. The periodic sequence region is defined and the periodic sequence units are queried for a multiple alignment to identify repetitive motif and its sequence logo. The representative motif sequence is aligned back to the sequence of the periodically repeating region to annotate the repeating units. Finally, the annotations for detected periodic repeats are added to the input information and output in GenBank format with an option of visualizing k-mer density, position-periodicity matrix, sequence motif logo and repetitive unit loci with neighboring genes for each periodic repeat.

Once the core regions have been extracted we can apply distance as well as clustering methods in order to discover any underlying structures. Towards this direction, there are many distance functions that compute text similarity. Our tool uses the *textdistance*[6] Python library which has implementations for more than thirty text distance algorithms. Furthermore, we also implemented the approach described in [1].

In this case, the nucleotide repeat string is cast in a twelve dimensional space. This is done by converting the repeat string into three strings according to the categories of nucleotide bases, and then yields a 12-dimension feature vector. The feature values are computed by an entropy based model that takes both local word frequency and position information into account [1].

5 Results

This section presents some initial TUF region mappings as these are visualised on the IGV browser. Concretely, we compute the Viterbi paths for chromosomes 1, 2. The Viterbi path simply answers the following question [8]; given an HMM λ and a sequence of observations O we seek to find the state sequence Q that maximizes the probability

$$P(Q|O,\lambda) \tag{10}$$

[5] See https://github.com/yachielab/SPADE for details

[6] See https://pypi.org/project/textdistance/ for more details.

We extract regions typically of size 20×10^6 bases. The regions are discretised into non-overlapping windows of size 100 bases. Each of the windows has a view of both samples, i.e. m605 and m585. The same cutoff mean that is used previously is also applied. This is the same approach we used in the previous section. However, in this section, Gap windows are included in the formed sequence. In the present context, the observations are pairs of RD means corresponding to the sample view that each window contains. The HMM model discussed in section 3 is used to compute the Viterbi path for the sequence. Overall, we plot the mappings for the following regions for chromosomes 1 and 2

- $[1 - 20] \times 10^6$
- $[20 - 40] \times 10^6$
- $[40 - 60] \times 10^6$
- $[60 - 80] \times 10^6$

Figures 10, 11 present the classification of the windows after applying the Viterbi algorithm for chromosome 1. Figures 12 and 13 shows the mappings for these regions as these are visualised on the IGV browser [9]. The red stripes correspond to the TUF regions so that they are easier to distinguish.

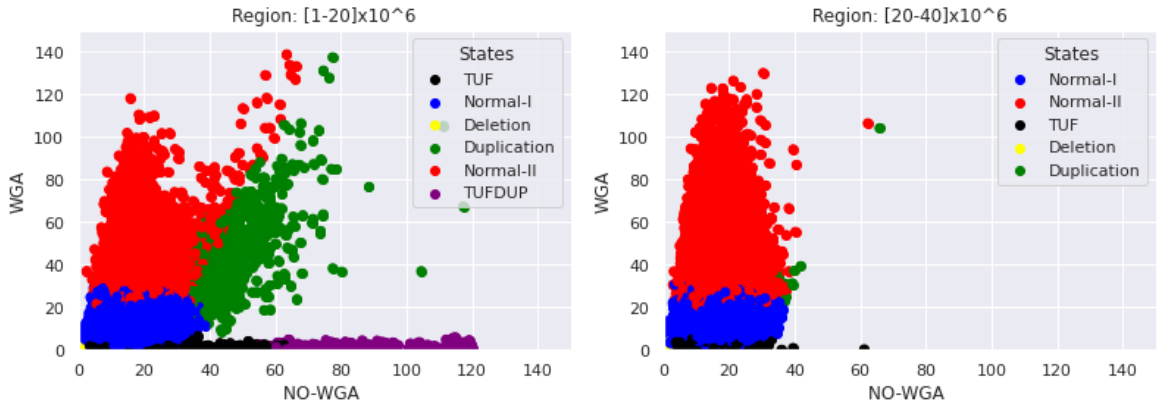

FIGURE 10: Regions 1 and 2 for chromosome 1. Left to right.

Similarly, figures 14 and 15 show the labelling of the windows with respect to the their mean value for regions 1, 2, 3 and for 4 for chromosome 2. Furthermore, figures 16 and 17 show the mapping of the regions in IGV browser.

Apart from region 1 for chromosome 1 the rest of the regions for both chromosomes are relatively uniform with respect to the mean RD count. This is illustrated both in the window labelling and in the IGV mapping. Note also that the GC count is shown somehow increased for region 1 of chromosome 1.

,

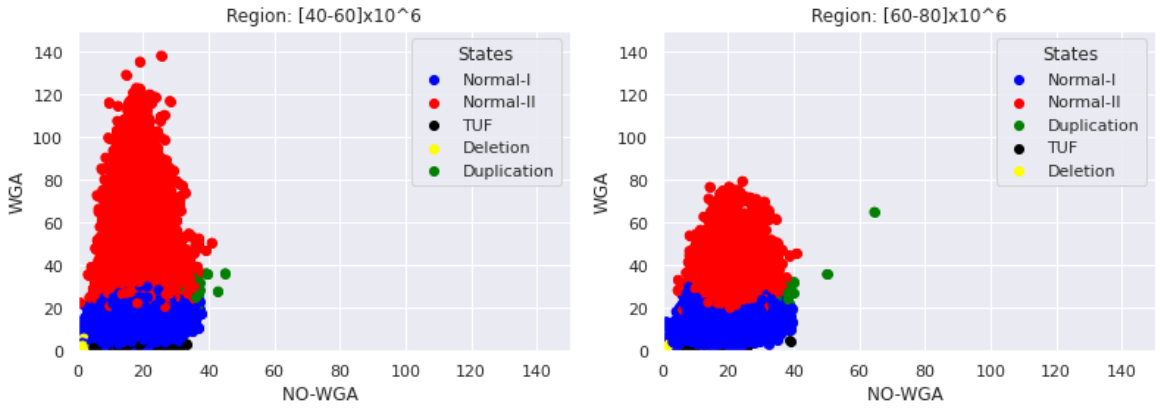

FIGURE 11: Regions 3 and 4 for chromosome 1. Left to right.

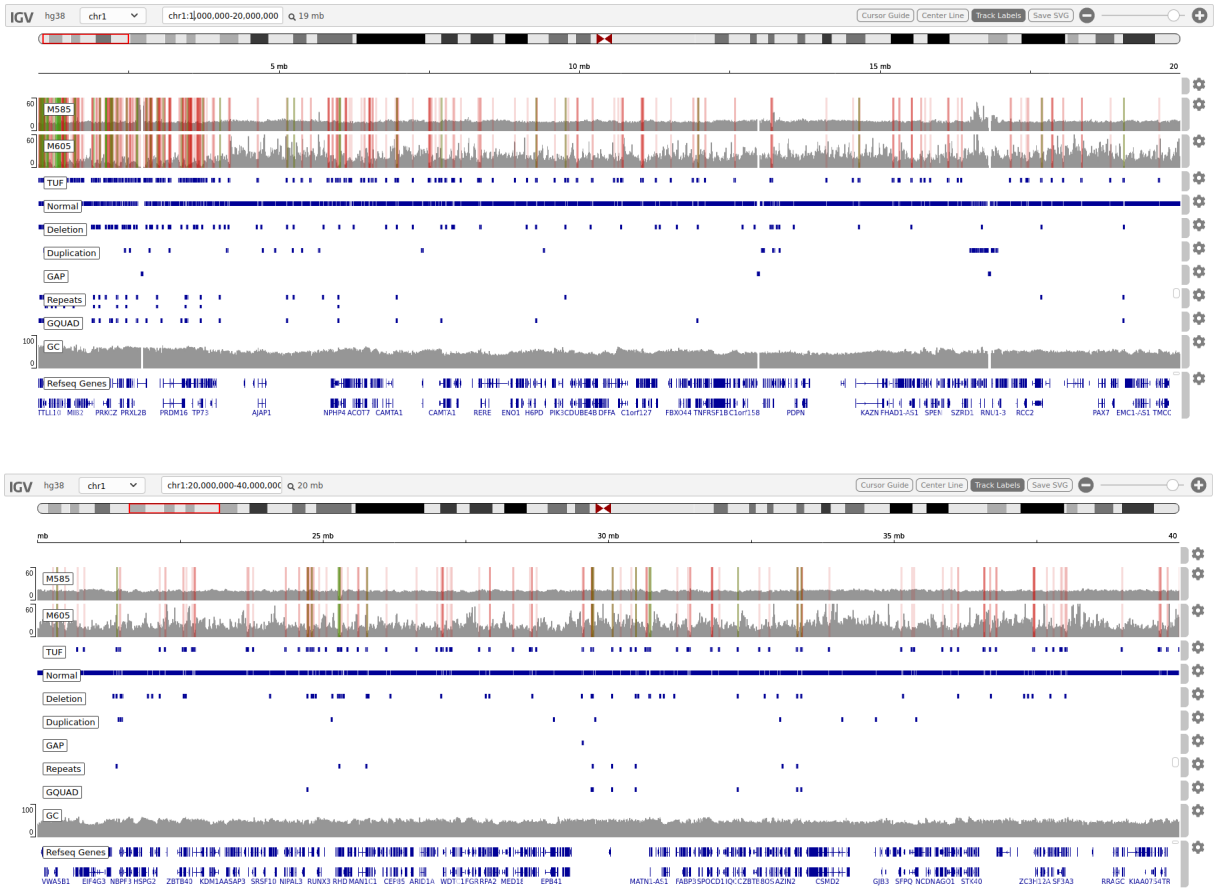

FIGURE 12: IGV mappings for regions 1 and 2 for chromosome 1.

,

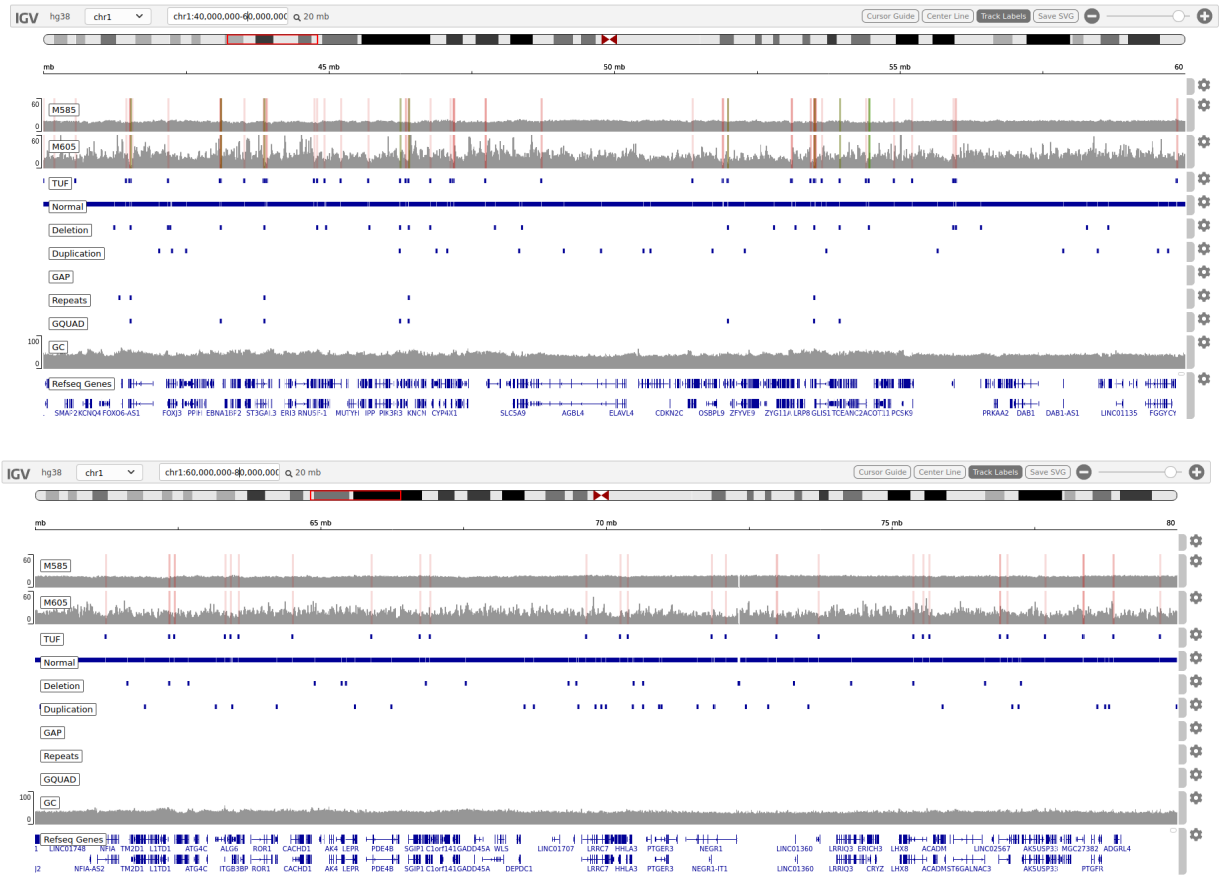

FIGURE 13: IGV mappings for regions 3 and 4 for chromosome 1.

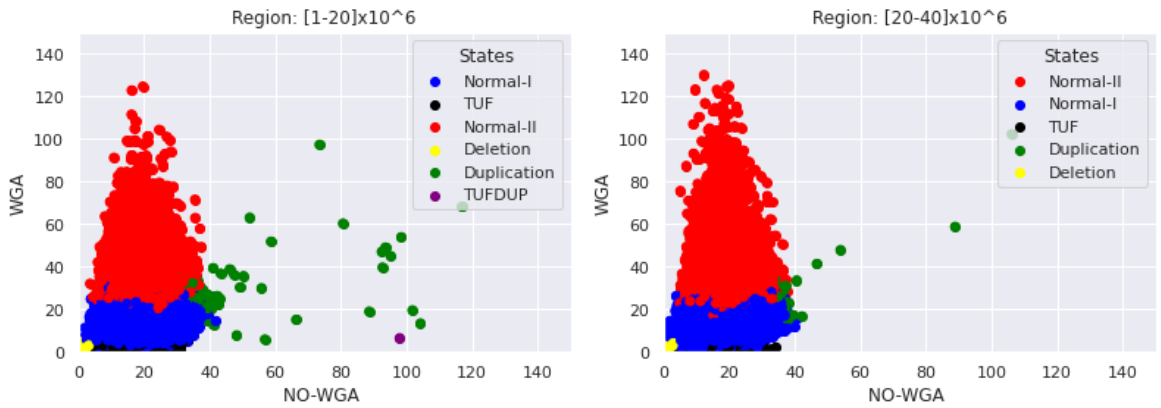

FIGURE 14: Regions 1 and 2 for chromosome 2. Left to right.

,

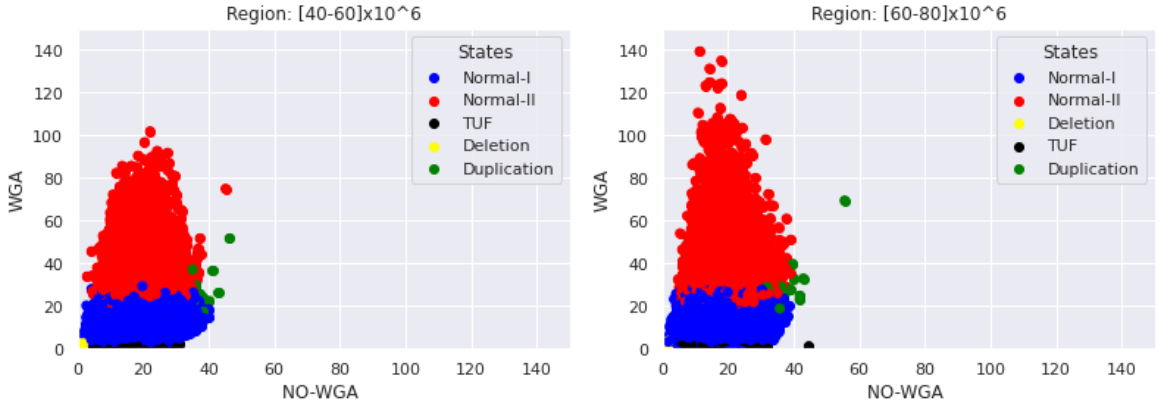

FIGURE 15: Regions 3 and 4 for chromosome 2. Left to right.

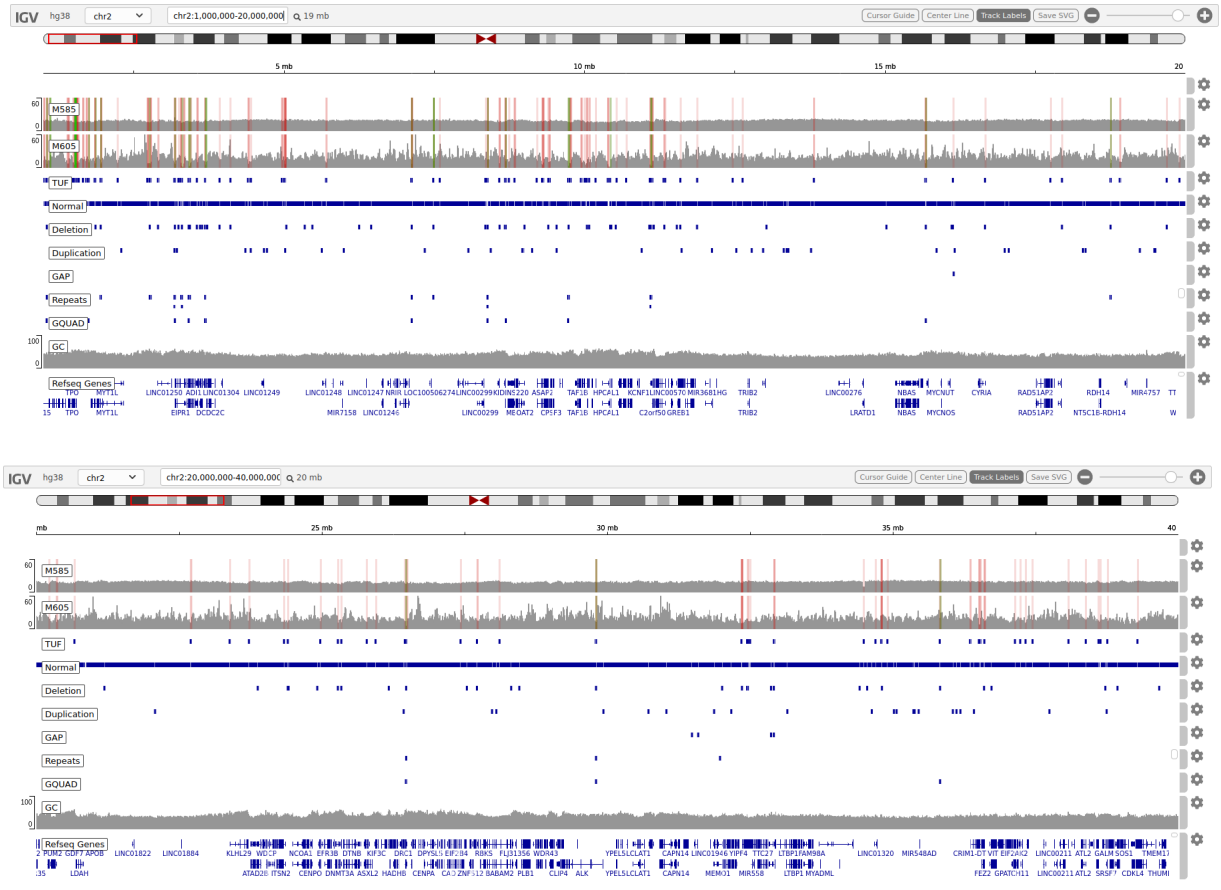

FIGURE 16: IGV mappings for regions 1 and 2 for chromosome 2.

,

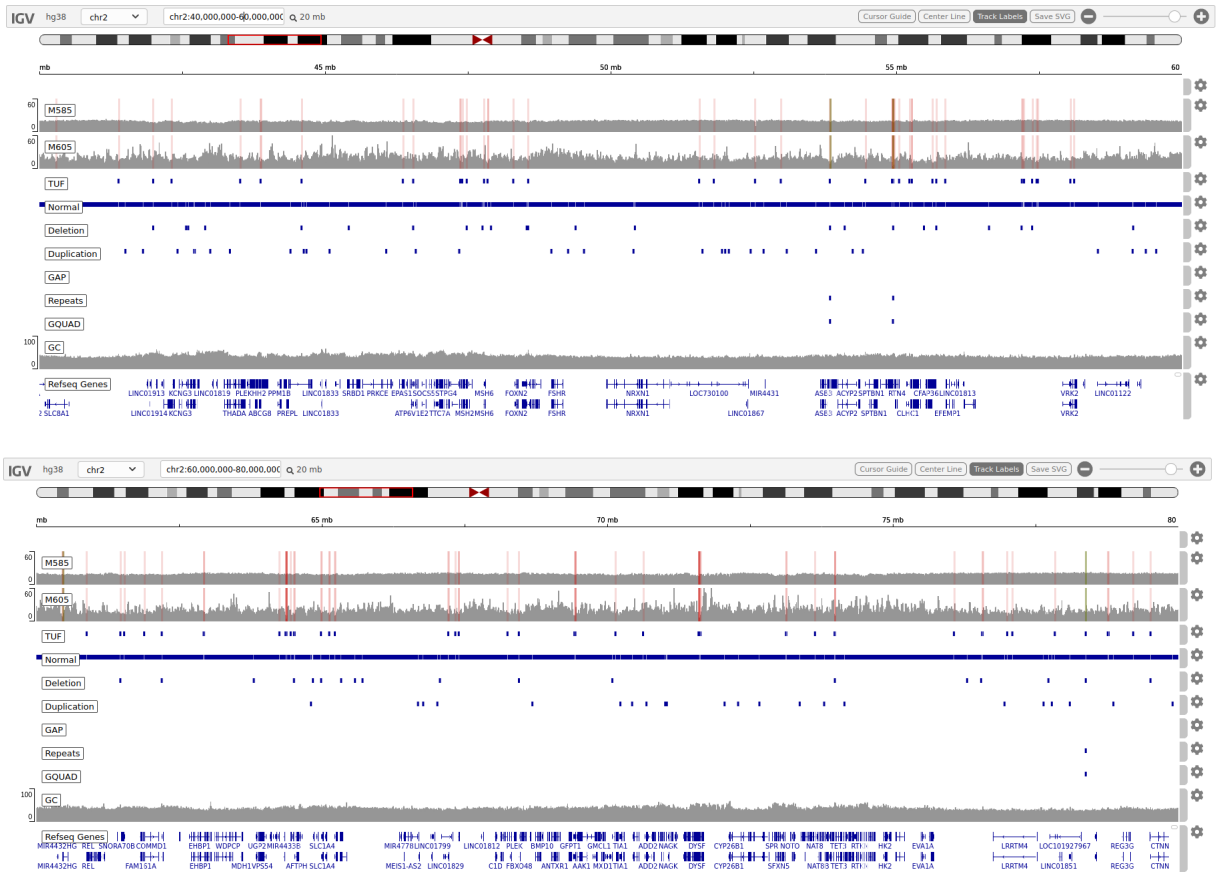

FIGURE 17: IGV mappings for regions 3 and 4 for chromosome 2.

6 Discussion

In this work, we describe the development of a hidden Markov model in order to map is described in order to map thermodynamically ultra-fastened DNA regions. These are stretches of the DNA which fail to denature even after the application of extreme melting conditions. The model uses two DNA sequences and assumes seven different states. Some early mappings are reported in order to show case the utilization of the approach. We have, so far, developed a tool where the user can import hidden Markov models and export Viterbi paths so given DNA sequences. Furthermore, the user can also request for the extraction of TUF core repeats.

Future work should include extensive validation of the model. This however may require a meticulous labelling of DNA regions. Once the model is validated, one can use it to extract the identified TUF regions and further investigate or model their structure.

,

References

[1] Junpeng Bao, Ruiyu Yuan, and Zhe Bao. An improved alignment-free model for dna sequence similarity metric. *BMC Bionformatics*, 15, 2014.

[2] Patric Cahan, Laura E. Godfrey, Peggy S. Eis, Todd A. Richmond, Rebecca R. Selzer, Michael Brent, Howard L. McLeod, Timothy J. Ley, and Timothy A. Graubert. wuhmm: a robust algorithm to detect dna copy number variation using long oligonucleotide microarray data. *Nuclic Acids Research*, 2008.

[3] Stefano Colella, Christopher Yau, Jennifer M. Taylor, Ghazala Mirza, Helen Butler, Penny Clouston, Anne S. Bassett, Anneke Seller, Christopher C. Holmes, and Jiannis Ragoussis. Quantisnp: an objective bayes hidden-markov model to detect and accurately map copy number variation using snp genotyping data. *Nuclic Acids Research*, 2007.

[4] P. Flach. *Machine Learning The art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[5] Jane Fridlyand, Antoine M. Snijders, Dan Pinkel, Donna G. Albertson, and Ajay N. Jain. Hidden markov models approach to the analysis of array cgh data. *Journal of Multivariate Analysis*, 2004.

[6] Jasmine Irani, Nitin Pise, and Madhura Phatak. Clustering techniques and the similarity measures used in clustering: A survey. *International Journal of Computer Applications*, 134, 2016.

[7] Tingting Liu and Jan Lemeire. Efficient and effective learning of hmms based on identification of hidden states. *Mathematical Problems in Engineering*, 2017.

[8] Rabiner L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Some Jurnal*, 2009.

[9] James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, and Jill P. Mesirov. Integrative genomics viewer. *Nature Biotechnology*, 2011.

[10] Alexander Schliep, Alexander Schönhuth, and Christine Steinhoff. Using hidden markov models to analyze gene expression time course data. *BIOINFORMATICS*, 2003.

[11] Koski T. *Hidden Markov Models for Bioinformatics*. Kluwer Academic Publishers, 2001.

[12] Colin D Veal, Peter J Freeman, Kevin Jacobs, Owen Lancaster, Stéphane Jamain, Marion Leboyer, Demetrius Albanes, Reshma R Vaghela, Ivo Gut, Stephen J Chanock, and Anthony J Brookes. A mechanistic basis for amplification differences between samples and between genome regions. *BMC Genomics*, 13, 2012.

[13] Kai Wang, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan F.A. Grant, Hakon Hakonarson, and Maja Bucan. Penncnv: An integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. *Genome Research*, 2007.