Fast Bayesian Inference of Copy Number Variants using Hidden Markov Models with Wavelet Compression

John Wiedenhoeft, Eric Brugel, Alexander Schliep
Department of Computer Science
Rutgers University
{john.wiedenhoeft, schliep}@cs.rutgers.edu

July 31, 2015

Abstract

By combining Haar wavelets with Bayesian Hidden Markov Models, we improve detection of genomic copy number variants (CNV) in array CGH experiments compared to the state-of-the-art, including standard Gibbs sampling. At the same time, we achieve drastically reduced running times, as the method concentrates computational effort on chromosomal segments which are difficult to call, by dynamically and adaptively recomputing consecutive blocks of observations likely to share a copy number. This makes routine diagnostic use and re-analysis of legacy data collections feasible; to this end, we also propose an effective automatic prior. An open source software implementation of our method is available at `http://bioinformatics.rutgers.edu/Software/HaMMLET/`. The web supplement is at `http://bioinformatics.rutgers.edu/Supplements/HaMMLET/`

Background

The human genome shows remarkable plasticity, leading to significant copy number variations (CNV) within the human population [1]. They contribute to differences in phenotype [2–4], ranging from benign variation over disease susceptibility to inherited or somatic diseases [5], including neuropsychiatric disorders [6–8] and cancer [9, 10]. Separating common from rare variants is important in the study of genetic diseases [5, 11, 12], and while the experimental platforms have matured, interpretation and assessment of pathogenic significance remains a challenge [13].

Computationally, CNV detection is a segmentation problem, in which consecutive stretches of the genome are to be labeled by their copy number. Along with a variety of other methods [14–40], Hidden Markov Models (HMM) [41] play a central role [42–52], as they directly model the separate layers of observed measurements, such as log-ratios in array CGH, and their corresponding latent copy number (CN) states, as well as the underlying linear structure of segments.

As statistical models, they depend on a large number of parameters, which have to be either provided *a priori* by the user or inferred from the data. Classic frequentist maximum likelihood (ML) techniques like Baum-Welch [53, 54] are not guaranteed to be globally optimal, i. e. they can converge to the wrong parameter values, which can limit the accuracy of the segmentation. Furthermore, the Viterbi algorithm [55] only yields a single maximum a posteriori (MAP) segmentation given a parameter estimate [56]. Failure to consider the full set of possible parameters precludes alternative interpretations of the data, and makes it very difficult to derive p-values or confidence intervals. Furthermore, these frequentist techniques have come under increased scrutiny in the scientific community.

Bayesian inference techniques for HMMs, in particular Forward-Backward Gibbs sampling [57, 58], provide an alternative for CNV detection as well [59–61]. Most importantly, they yield a complete probability distribution of copy numbers for each observation. As they are sampling-based, they are computationally expensive, which is problematic especially for high-resolution data. Though they are guaranteed to converge to the correct values under very mild assumptions, they tend to do so slowly, which can lead to oversegmentation and mislabeling if the sampler is stopped prematurely.

Another issue in practice is the requirement to specify hyperparameters for the prior distributions. Despite the theoretical advantage of making the inductive bias more explicit, this can be a major source of annoyance for the user. It is also hard to justify any choice of hyperparameters when insufficient domain knowledge is available.

Recent work of our group [62] has focused on accelerating Forward-Backward Gibbs sampling through the introduction of compressed HMMs and approximate sampling. For the first time, Bayesian inference could be performed at running times on par with classic maximum likelihood approaches. It was based on a greedy spatial clustering heuristic, which yielded a static compression of the data into blocks, and block-wise sampling. Despite its success, several important issues remain to be addressed. The blocks are fixed throughout the sampling and impose a structure that turns out to be too rigid in the presence of variances differing between

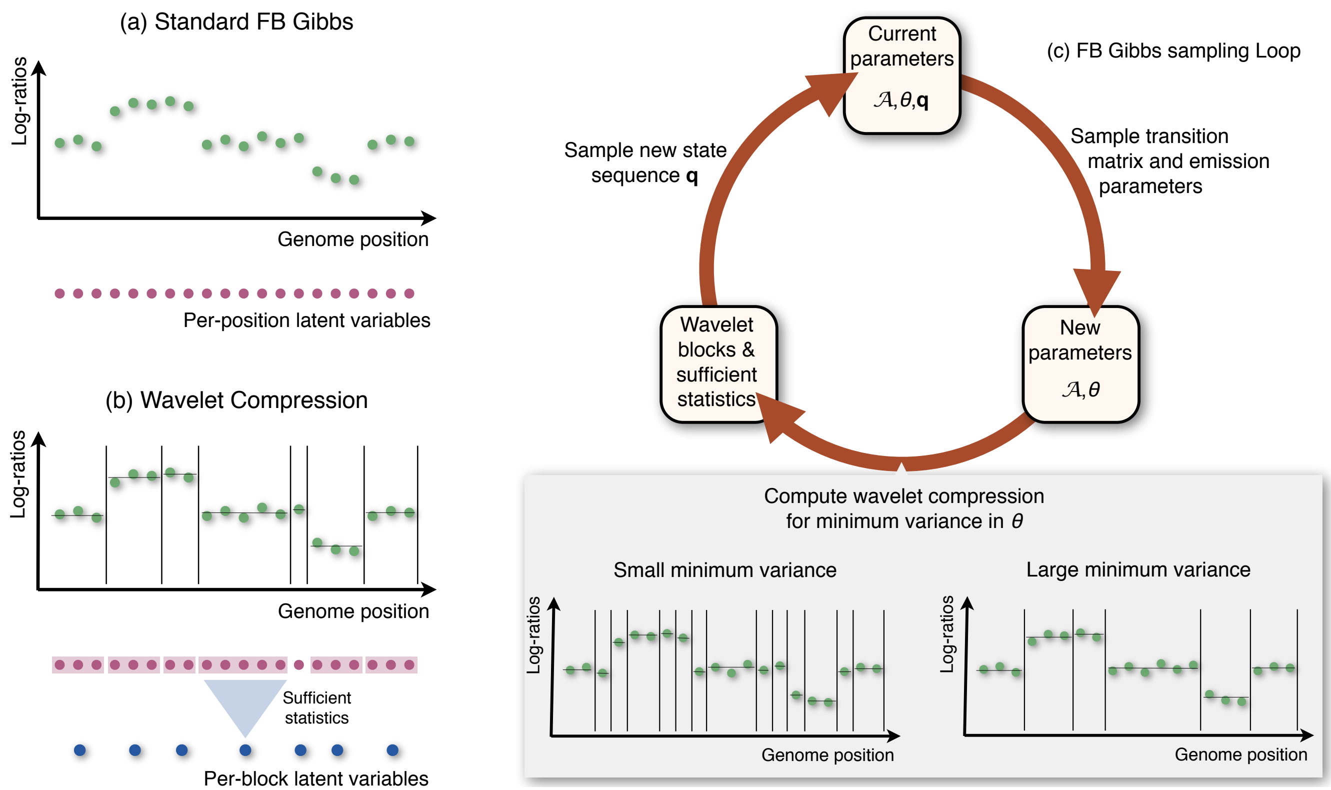

Figure 1. Overview of HaMMLET. Instead of individual computations per observation (panel a), Forward-Backward Gibbs Sampling is performed on a compressed version of the data, using sufficient statistics for block-wise computations (panel b) to accelerate inference in Bayesian Hidden Markov Models. During the sampling (panel c) parameters and copy number sequences are sampled iteratively. During each iteration, the sampled variances determine which coefficients of the data's Haar wavelet transform are dynamically set to zero. This controls potential break points at finer or coarser resolution or, equivalently, defines blocks of variable number and size (panel c, bottom). Our approach thus yields a dynamic, adaptive compression scheme which greatly improves speed of convergence, accuracy and running times.

CN states. The clustering heuristic relies on empirically derived parameters not supported by a theoretical analysis, which can lead to suboptimal clustering or overfitting. Also, the method cannot easily be generalized for multivariate data. Lastly, the implementation was primarily aimed at comparative analysis between the frequentist and Bayesian approach, as opposed to overall speed.

To address these issues and make Bayesian CNV inference feasible even on a laptop, we propose the combination of HMMs with another popular signal processing technology: Haar wavelets have previously been used in CNV detection [63], mostly as a preprocessing tool for statistical downstream applications [24–28] or simply as a visual aid in GUI applications [17, 64]. A combination of smoothing and segmentation has been suggested as likely to improve results [65], and here we show that this is indeed the case. The wavelets provide a theoretical foundation for a better, dynamic compression scheme for faster convergence and accelerated Bayesian sampling. We improve simultaneously upon the typically conflicting goals of accuracy and speed, because the wavelets allow summary treatment of "easy" CN calls in segments and focus computational effort on the "difficult" CN calls, dynamically and adaptively. This is in contrast to other computationally efficient tools, which often simplify the statistical model or use heuristics. The required data structure can be efficiently computed, incurs minimal overhead, and has a straightforward generalization to multivariate data. We further show how the wavelet transform yields a natural way to set hyperparameters automatically, with little input from the user.

We implemented our method in a highly optimized end-user software, called HaMMLET. Aside from achieving an acceleration of up to two orders of magnitude, it exhibits significantly improved convergence behavior, has excellent precision and recall, and provides Bayesian inference within seconds even for large data sets. The accuracy and speed of HaMMLET also makes it an excellent choice for routine diagnostic use and large-scale re-analysis of legacy data.

Results and discussion

Simulated aCGH data

A previous survey [65] of eleven CNV calling methods for aCGH has established that segmentation-focused methods such as DNAcopy [32, 33], an implementation of circular binary segmentation (CBS), as well as CGHseg [35] perform consistently well. DNAcopy performs a number of t-tests to detect break-point candidates. The result is typically over-segmented and requires a merging step in post-processing, especially to reduce the number of segment means. To this end MergeLevels was introduced by [66]. They compare the combination DNAcopy+MergeLevels to their own HMM implementation [45] as well as GLAD [23], showing its superior performance over both methods. This established DNAcopy+MergeLevels as the *de facto* standard in CNV

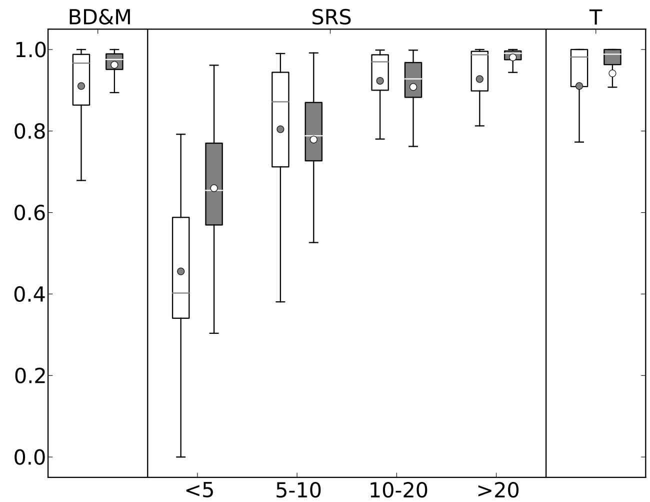

Figure 2. F-measures of CBS (light) and HaMMLET (dark) for calling aberrant copy numbers on simulate aCGH data. Boxes represent the interquartile range (IQR = Q3-Q1), with a horizontal line showing the median (Q2), whiskers representing the range ($\frac{3}{2}$IQR beyond Q1 and Q3), and the bullet representing the mean. HaMMLET has the same or better F-measures in most cases, and converges to 1 for larger segments, whereas CBS plateaus for aberrations greater than 10.

detection, despite the comparatively long running time.

The paper also includes aCGH simulations deemed to be reasonably realistic by the community. DNACopy was used to segment 145 unpublished samples of breast cancer data, and subsequently labeled as copy numbers 0 to 5 by sorting them into bins with boundaries $(-\infty, -0.4, -0.2, 0.2, 0.4, 0.6, \infty)$, based on the sample mean in each segment (the last bin appears to not be used). Empirical length distributions were derived, from which the sizes of CN aberrations are drawn. The data itself is modeled to include Gaussian noise, which has been established as a sufficient for aCGH data [67]. Means were generated such as to mimic random tumor cell proportions, and random variances were chosen to simulate experimenter bias often observed in real data; this emphasizes the importance of having automatic priors available when using Bayesian methods, as the means and variances might be unknown *a priori*. The simulations are available at `http://www.cbs.dtu.dk/~hanni/aCGH/`. They comprise three sets of simulations: "breakpoint detection and merging" (BD&M), "spatial resolution study" (SRS), and "testing" (T) (see their paper for details). We used the MergeLevels implementation as provided on their website. It should be noted that the superiority of DNAcopy+MergeLevels was established using a simulation based upon segmentation results of DNAcopy itself.

We used the Bioconductor package DNAcopy (version 1.24.0), and followed the procedure suggested therein, including outlier smoothing. This version uses the linear-time variety of CBS [34]; note that other authors such as [31] compare against its quadratic-time version [33], which is likely to yield a favorable comparison and speedups of several orders of magnitude, especially on large data. For HaMMLET, we use a 5-state model with automatic hyperparameters $\mathbb{P}(\sigma^2 \leq 0.01) = 0.9$ (see section Automatic priors), and all Dirichlet hyperparameters set to 1.

We report F-measures for detecting aberrant segments (Fig. 2). On datasets T and BD&M, both methods have similar medians, but HaMMLET has a much better interquartile range (IQR) and range, about half of CBS's. On the spatial resolution data set (SRS), HaMMLET performs much better on very small aberrations. This might seem somewhat surprising, as short segments could easily get lost under compression. However, Lai *et al.* [65] have noted that smoothing-based methods such as quantile smoothing (quantreg) [19], lowess [20], and wavelet smoothing [25] perform particularly well in the presence of high noise and small CN aberrations, suggesting that "an optimal combination of the smoothing step and the segmentation step may result in improved performance". Our wavelet-based compression inherits those properties. For CNVs of sizes between 5 and 10, CBS and HaMMLET have similar ranges, with CBS being skewed towards better values; CBS has a slightly higher median for 10–20, with IQR and range being about the same. However, while HaMMLET's F-measure consistently approaches 1 for larger aberrations, CBS does not appear to significantly improve after size 10.

High-density CGH array

In this section, we demonstrate HaMMLET's performance on biological data. Due to the lack of a gold standard for high-resolution platforms, we assess the CNV calls qualitatively. We use raw aCGH data (GEO:GSE23949) [68] of genomic DNA from breast cancer cell line BT-474 (invasive ductal carcinoma, GEO:GSM590105), on an Agilent-021529 Human CGH Whole Genome Microarray 1x1M platform (GEO:GPL8736). We excluded gonosomes, mitochondrial and random chromosomes from the data, leaving 966,432 probes in total.

HaMMLET allows for using automatic emission priors (see section Automatic priors) by specifying a noise variance, and a probability to sample a variance not exceeding this value. We compare HaMMLET's performance against CBS, using a 20-state model with automatic priors, $\mathbb{P}(\sigma^2 \leq 0.1) = 0.8$, 10 prior self-transitions and 1 for all other hyperparameters. CBS took over 2 h 9 m to process the entire array, whereas HaMMLET took 27.1 s for 100 iterations, a speedup of 288. The compression ratio (see section Speed and convergence effects of wavelet compression) was 220.3. CBS yielded a massive over-segmentation into 1,548 different copy number levels. As the data is derived from a relatively homogeneous cell line as opposed to a biopsy, we do not expect the presence of subclonal populations to be a contributing factor [69, 70]. Instead, measurements on aCGH are known to be spatially correlated, resulting in a wave pattern which has to be removed in a preprocessing step. CBS performs such a smoothing, yet the unrealistically large number of different levels is likely due to residuals of said wave pattern. Repeated runs of CBS yielded different numbers of levels, suggesting that indeed the merging was incomplete. This further highlights the importance of prior knowledge in statistical inference. By limiting the number of CN states *a priori*, non-i.i.d. effects are easily remedied, and inference can be performed without preprocessing. Furthermore, the internal compression mechanism of HaMMLET is derived from a spatially adaptive regression method, so smoothing is inherent to our method. The results are included in the supplement.

For a more comprehensive analysis, we restricted our evaluation to chromosome 20 (21,687 probes), which we assessed to be the most complicated to infer, as it appears to have the highest number of different CN states and breakpoints. CBS yields a 19-state result after 15.78 s (Fig. 3, top). We have then used a 19-state model automated priors ($\mathbb{P}(\sigma^2 \leq 0.04) = 0.9$), 10 prior self-transitions, all other Dirichlet parameters set to 1) to reproduce this result. Using noise control (see Methods), our method took 1.61 s for 600 iterations. The solution we obtained is consistent with CBS (Fig. 3, middle and bottom). However, only 11 states were part of the final solution, i. e. 6 states yielded no significant likelihood above that of other states. We observe superfluous states being ignored in our simulations as well, cf. Supplement. In light of the results on the entire array, we suggest that the segmentation by DNAcopy has not sufficiently been merged by MergeLevels. Most strikingly, HaMMLET does not show any marginal support for a segment called by CBS around probe number 4,500. We have confirmed that this is not due to data compression, as the segment is broken up into multiple blocks in each iteration (data not shown). On the other hand, two much smaller segments called by CBS in the 17,000–20,000 range do have marginal support of about 40% in HaMMLET, suggesting that the lack of support for the larger segment is correct. It should be noted that inference differs between the entire array and chromosome 20 in both methods, since long-range effects have higher impact in larger data.

We also demonstrate another feature of HaMMLET called *noise control*. While Gaussian emissions have been deemed a sufficiently accurate noise model for aCGH [67], microarray data is prone to outliers, for example due to damages on the chip. While it is possible to model outliers directly [60], the characteristics of the wavelet transform allow us to largely suppress them during the construction of our data structure (see Methods). Notice that due to noise control most outliers are correctly labeled according to the segment they occur in, while the short gain segment close to the beginning is called correctly.

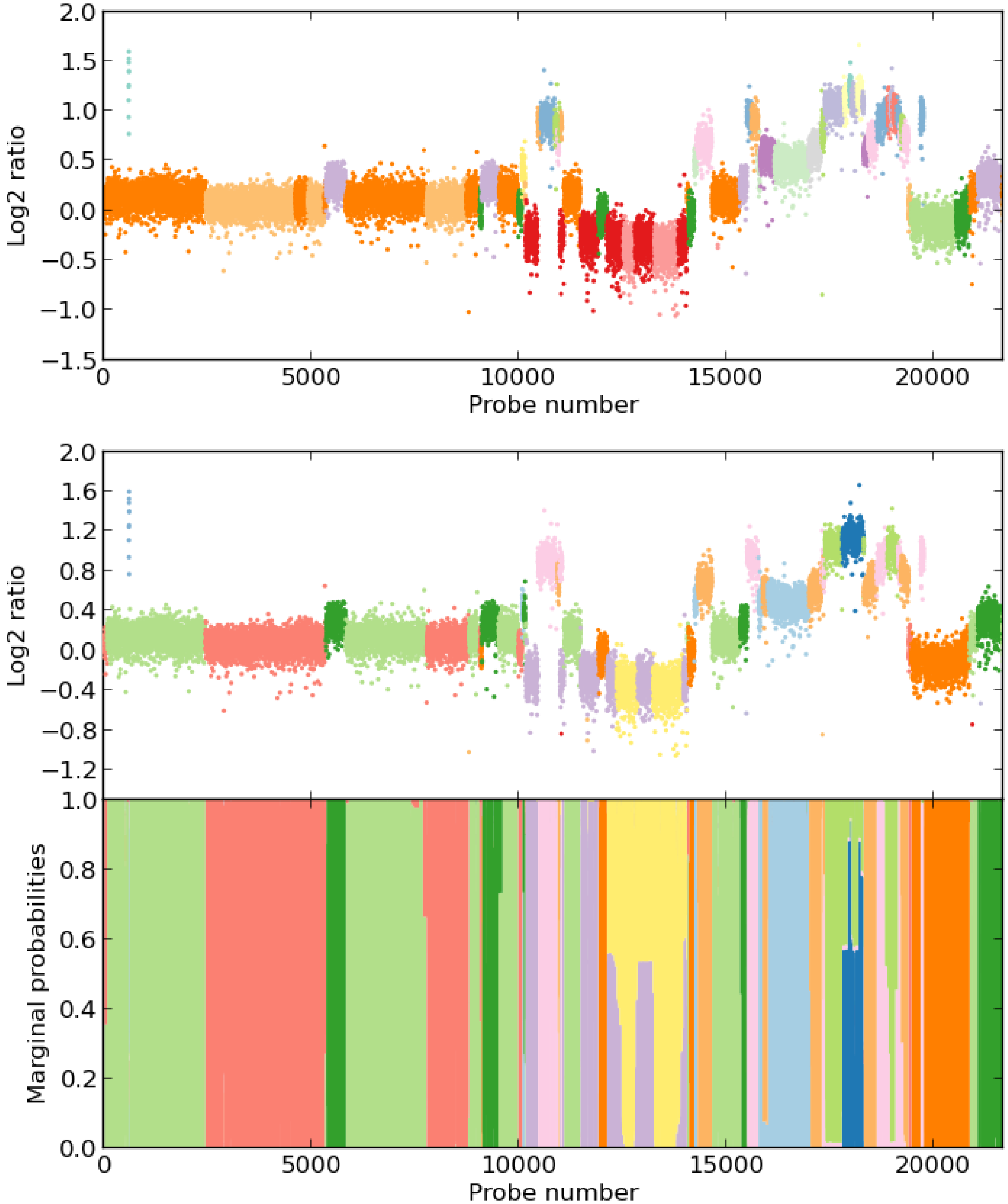

Figure 3. Copy number inference for chromosome 20 in invasive ductal carcinoma (21,687 probes). CBS creates a 19-state solution (top), however, a compressed 19-state HMM only supports an 11-state solution (bottom), suggesting insufficient level merging in CBS.

Speed and convergence effects of wavelet compression

The speedup gained by compression depends on how well the data can be compressed. Poor compression is expected when the means are not well separated, or short segments have small variance, which necessitates the creation of smaller blocks for the rest of the data to expose potential low-variance segments to the sampler. On the other hand, data must not be over-compressed to avoid merging small aberrations with normal segments, which would decrease the F-measure. The level of compression is measured with the compression ratio. Due to the dynamic changes to the block structure, we define compression ratio as the total number of blocks in all iterations, divided by the product of the number of data points T and the number of iterations N. As usual a compression ratio of one indicates no compression.

To evaluate the impact of dynamic wavelet compression on speed and convergence properties of an HMM, we created 129,600 different data sets with $T = 32{,}768$ many probes. In each data set, we randomly distributed 1 to 6 gains of a total length of $\{100, 250, 500, 750, 1000\}$ uniformly among the data, and do the same for losses. Mean combinations $(\mu_{\text{loss}}, \mu_{\text{neutral}}, \mu_{\text{gain}})$ were chosen from $(\log_2 \frac{1}{2}, \log_2 1, \log_2 \frac{3}{2})$, $(-1, 0, 1)$, $(-2, 0, 2)$, and $(-10, 0, 10)$, and variances $(\sigma^2_{\text{loss}}, \sigma^2_{\text{neutral}}, \sigma^2_{\text{gain}})$ from $(0.05, 0.05, 0.05)$, $(0.5, 0.1, 0.9)$, $(0.3, 0.2, 0.1)$, $(0.2, 0.1, 0.3)$, $(0.1, 0.3, 0.2)$, and $(0.1, 0.1, 0.1)$. These values have been selected to yield a wide range of easy and hard cases, both well separated, low-variance data with large aberrant segments as well as cases in which small aberrations overlap significantly with the tail samples of high-variance neutral segments. Consequently, compression ratios range from $\sim$1 to $\sim$2,100. We use automatic priors $\mathbb{P}(\sigma^2 \leq 0.2) = 0.9$, self-transition priors $\alpha_{ii} \in \{10, 100, 1000\}$, non-self transition priors $\alpha_{ij} = 1$, and initial state priors $\boldsymbol{\alpha} \in \{\mathbf{1}, \mathbf{10}\}$. Using all possible combinations of the above yields 129,600 different simulated data sets, a total of 4.2 billion values.

We achieve speedups per iteration of up to 350 compared to an uncompressed HMM (Fig. 5). In contrast, [62] have reported ratios of 10–60, with one instance of 90. Notice that the speedup is not linear in the compression ratio. While sampling itself is expected to yield linear speedup, the marginal counts still have to be tallied individually for each position, and dynamic block creation causes some overhead. The quantization artifacts observed for larger speedup are likely due to the limited resolution of the Linux time command (10 milliseconds). Compressed HaMMLET took about 11.2 CPU hours for all 129,600 simulations, whereas the uncompressed version took over 3 weeks and 5 days. All running times reported are CPU time measured on a single core of a AMD Opteron™ 6174 Processor, clocked at 2.2 GHz.

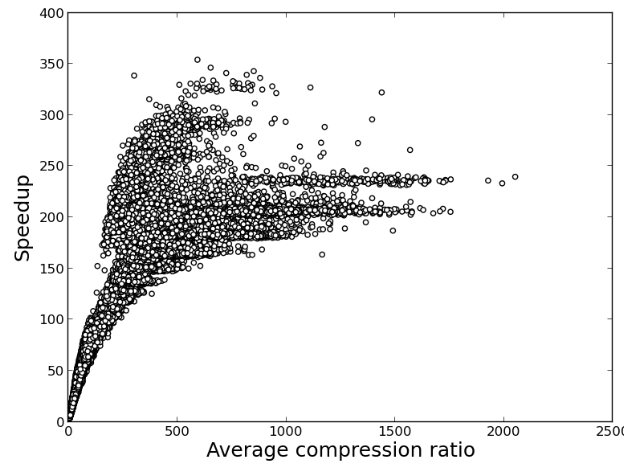

Figure 5. HaMMLET's speedup as a function of the average compression during sampling. As expected, higher compression leads to greater speedup. The non-linear characteristic is due to the fact that some overhead is incurred by the dynamic compression, as well as parts of the implementation that do not depend on the compression, such as tallying marginal counts.

We evaluate the convergence of the F-measure of compressed and uncompressed inference for each simulation. Since we are dealing with multi-class classification, we use the micro- and macro-averaged F-measures (F_{mi}, F_{ma}) proposed by [71]; they tend to be dominated by the classifier's performance on common and rare categories, respectively. Since all state labels are sampled from the same prior and hence their relative order is random, we used the label permutation which yielded the highest sum of micro- and macro-averaged F-measures.

In Fig. 4, we show that the compressed version of the Gibbs sampler converges almost instantly, whereas the uncompressed version converges much slower, with about 5% of the cases failing to yield an F-measure > 0.6 within 1,000 iterations. Wavelet compression is likely to yield reasonably large blocks for the majority class early on, which leads to a strong posterior estimate of its parameters and self-transition probabilities. As expected, F_{ma} are generally worse, since any misclassification in a rare class has a larger impact. Especially in the uncompressed version, we observe that F_{ma} tends to plateau until F_{mi} approaches 1.0. Since any misclassification in the majority (neutral) class adds false positives to the minority classes, this effect is expected. It implies that correct labeling of the majority class is a necessary condition for correct labeling of minority classes, in other words, correct identification of the rare, interesting segments requires the sampler to properly converge, which is much harder to achieve without compression. It should be noted that running compressed HaMMLET for 1,000 iterations is unnecessary on the simulated data, as in all cases it converges between 25 and 50 iterations. Thus, for all practical purposes, an additional speedup of 40–80 can be achieved by reducing the number of iterations, which yields convergence up to 3 orders of magnitude faster than standard FBG.

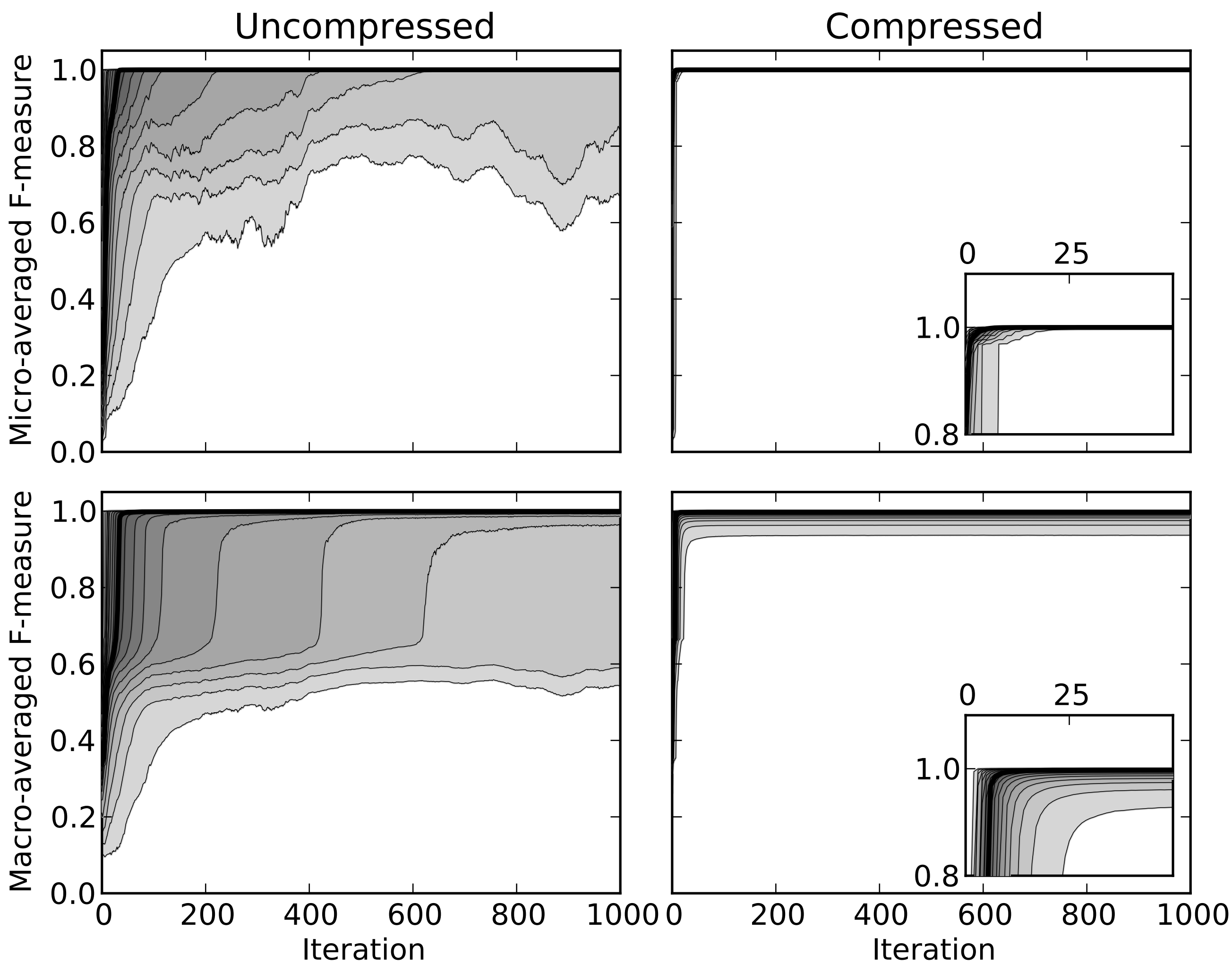

Figure 4. The median value (black) and quantile ranges (in 5% steps) of the micro- (top) and macro-averaged (bottom) F-measures for uncompressed (left) and compressed (right) FBG inference, on the same 129,600 simulated data sets, using automatic priors. The x-axis represents the number of iterations alone, and does not reflect the additional speedup obtained through compression. Notice that the compressed HMM converges no later than 50 iterations (inset figures, right).

Coriell, ATCC and breast carcinoma

The data provided by [72] includes 15 aCGH samples for the Coriell cell line. At about 2,000 probes, the data is small compared to modern high-density arrays. Nevertheless, these data sets have become a common standard to evaluate CNV calling methods, as they contain few and simple aberrations. The data also contains 6 ATCC cell lines as well as 4 breast carcinoma, all of which are substantially more complicated, and typically not used in software evaluations. In Fig. 6, we demonstrate our ability to infer the correct segments on the most complex example, a T47D breast ductal carcinoma sample of a 54 year old female. We used 6-state automatic priors with $\mathbb{P}(\sigma^2 \leq 0.1) = 0.85$, and all Dirichlet hyperparameters set to 1. On a standard laptop, HaMMLET took 0.09 seconds for 1,000 iterations; running times for the other samples were similar. Our results for all 25 data sets have been included in the supplement.

Conclusions

In the analysis of biological data, there are usually conflicting objectives at play which need to be balanced: the required accuracy of the analysis, ease of use—using the software, setting software and method parameters—and often the speed of a method. Bayesian methods have obtained a reputation of requiring enormous computational effort and being difficult to use, for the expert knowledge required for choosing prior distributions. It has also been recognized [60, 62, 73] that they are very powerful and accurate, leading to improved, high-quality results and providing, in the form of posterior distributions, an accurate measure of uncertainty in results. Nevertheless, it is not surprising that a hundred times larger effort in computational effort alone prevented wide-spread use.

Inferring Copy Number Variants (CNV) is a quite special problem, as experts can identify CN changes visually, at least on very good data and for large, drastic CN changes (e. g., long segments lost on both chromosomal copies). With lesser quality data, smaller CN differences and in the analysis of cohorts the need for objective, highly accurate, and automated methods are evident.

The core idea for our method expands on our prior work [62] and affirms a conjecture by Lai *et al.* [65] that a combination of smoothing and segmentation will likely improve results. One ingredient of our method are Haar wavelets, which have previously been used for pre-processing and visualization [17, 64]. In a sense, they quantify and identify regions of high variation and allow to summarize the data at various levels of resolution, somewhat similar to how an expert would perform a visual analysis. We combine, for the first time, wavelets with a full Bayesian HMM by dynamically and adaptively infering blocks of subsequent observations from our wavelet data structure. The HMM operates on blocks instead of individual observations which leads to great saving in running times, up to 350-fold compared to the standard FB-Gibbs sampler, and up to 288 times faster than CBS. Much more importantly, operating on the blocks greatly improves convergence of the sampler, leading to higher accuracy for a much smaller number of sampling iterations. Thus, the combination of wavelets and HMM realizes a simultaneous improvement on accuracy and on speed; typically one can have one or the other. An intuitive explanation as to why this works is that the blocks derived from the wavelet structure allow efficient, summary treatment of those “easy” to call segments given the current sample of HMM parameters and identifies “hard” to call CN segment which need the full computational effort from FB-Gibbs. Note that it is absolutely crucial that the block structure depends on the parameters sampled for the HMM and will change drastically over the run time. This is in contrast to our prior work [62], which used static blocks and saw no improvements to accuracy and convergence speed. The data structures and linear-time algorithms we introduce here provide the efficient means for recomputing these blocks at every cycle of the sampling, cf. Fig. 1. Compared to our prior work, we observe a speed-up of up to 3,000 due to the greatly improved convergence, $O(T)$ vs. $O(T \log T)$ clustering, improved numerics and, lastly, a C++ instead of a Python implementation.

We performed an extensive comparison with the state-of-the-art as identified by several review and benchmark publications and with the standard FB-Gibbs sampler on a wide range of biological data sets and 129,600 simulated data sets, which were produced by a simulation process not based on HMM to make it a harder problem for our method. All comparisons demonstrated favorable results for our method when measuring accuracy at a very noticeable acceleration compared to the state-of-the-art. It must be stressed that these results were obtained with a statistically sophisticated model for CNV calls and without cutting algorithmic corners, but rather an effective allocation of computational effort.

All our computations are performed using our automatic prior, which derives estimates for the hyperparameters of the priors for means and variances directly from the wavelet tree structure and the resulting blocks. The block structure also imposes a prior on self-transition probabilities. The user only has to provide an estimate of the noise variance, but as the automatic prior is designed to be weak the prior and thus the method is robust against incorrect estimates. We have demonstrated this by using different hyperparameters for the associated Dirichlet priors in our simulations, which HaMMLET is able to infer correctly regardless of the transition priors. At the same time the automatic prior can be used to tune certain aspects of the HMM if stronger prior knowledge is available. We would expect further improvements from non-automatic, expert-selected priors, but refrained from using them for the evaluation, as they might be perceived as unfair to other methods.

In summary, our method is as easy to use as other, statistically less sophisticated tools, more accurate and much

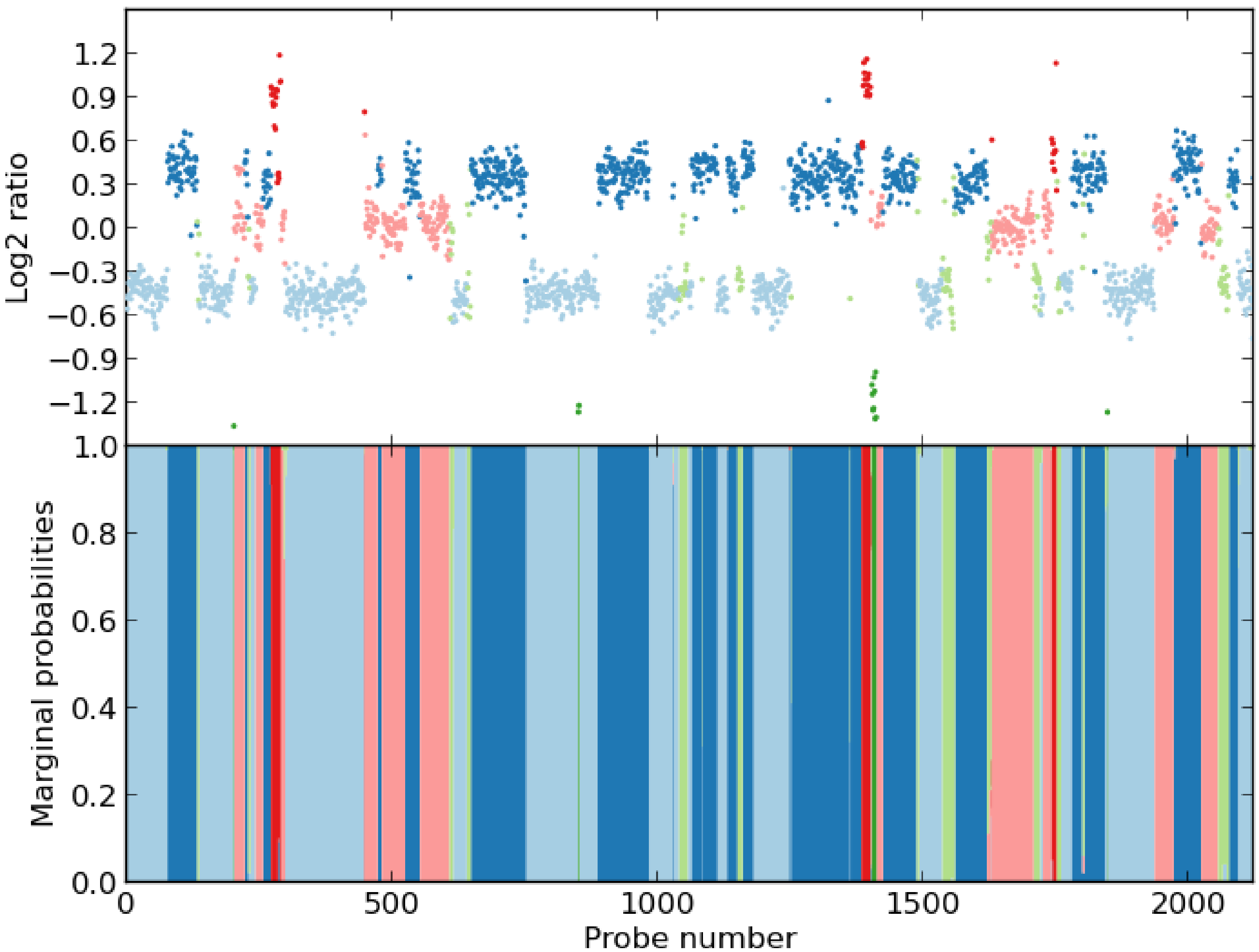

Figure 6. HaMMLET's inference of copy-number segments on T47D breast ductal carcinoma. Notice that the data is much more complex than the simple structure of a diploid majority class with some small aberrations typically observed for Coriell data.

more computationally efficient. We believe this makes it an excellent choice both for routine use in clinical settings and principled re-analysis of large cohorts, where the added accuracy and the much improved information about uncertainty in copy number calls from posterior marginal distributions will likely yield improved insights into CNV as a source of genetic variation and its relationship to disease.

Methods

We will briefly review Forward-Backward Gibbs sampling (FBG) for Bayesian Hidden Markov Models, and its acceleration through compression of the data into blocks. By first considering the case of equal emission variances among all states, we show that optimal compression is equivalent to a concept called *selective wavelet reconstruction*, following a classic proof in wavelet theory. We then argue that *wavelet coefficient thresholding*, a variance-dependent minimax estimator, allows for compression even in the case of unequal emission variances. This allows the compression of the data to be adapted to the prior variance level at each sampling iteration. We then derive a simple data structure to dynamically create blocks with little overhead. While wavelet approaches have been used for aCGH data before [25, 29, 30, 63], our method provides the first combination of wavelets and HMMs.

Bayesian Hidden Markov Models

Let T be the length of the observation sequence, which is equal to the number of probes. An HMM can be represented as a statistical model $(\mathbf{q}, \mathscr{A}, \theta \,|\, \mathbf{y})$, with transition matrix $\mathscr{A}$, a latent state sequence $\mathbf{q} = (q_0, q_1, \ldots, q_{T-1})$, an observed emission sequence $\mathbf{y} = (y_0, y_1, \ldots, y_{T-1})$, and the emission parameters θ. The initial distribution of q_0 is often included and denoted as π, but in a Bayesian setting it makes more sense to take π as a set of hyperparameters for q_0.

In the usual frequentist approach, the state sequence $\mathbf{q}$ is inferred by first finding a maximum likelihood estimate of the parameters,

$$(\mathscr{A}_{\mathrm{ML}}, \theta_{\mathrm{ML}}) = \arg\max_{(\mathscr{A},\theta)} \mathscr{L}(\mathscr{A}, \theta \,|\, \mathbf{y}),$$

using the Baum-Welsh algorithm [53, 54]. This is only guaranteed to yield local optima, as the likelihood function is not convex. Repeated random reinitializations are used to find "good" local optima, but there are no guarantees for this method. Then, the most likely state sequence given those parameters,

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} \mathbb{P}(\mathbf{q} \,|\, \mathscr{A}_{\mathrm{ML}}, \theta_{\mathrm{ML}}, \mathbf{y}),$$

is calculated using the Viterbi algorithm [55]. However, if there are only a few aberrations, that is there is imbalance between classes, the ML parameters tend to overfit the normal state which is likely to yield incorrect segmentation [62]. Furthermore, alternative segmentations given those parameters are also ignored, as are the ones for alternative parameters.

The Bayesian approach is to calculate the distribution of state sequences directly by integrating out the emission and transition variables,

$$\mathbb{P}(\mathbf{q} \,|\, \mathbf{y}) = \int_{\mathscr{A}} \int_{\theta} \mathbb{P}(\mathbf{q}, \mathscr{A}, \theta \,|\, \mathbf{y})\, \mathrm{d}\theta \, \mathrm{d}\mathscr{A}.$$

Since this integral is intractable, it has to be approximated using Markov Chain Monte Carlo techniques, i. e. drawing N samples,

$$(\mathbf{q}^{(i)}, \mathscr{A}^{(i)}, \theta^{(i)}) \sim \mathbb{P}(\mathbf{q}, \mathscr{A}, \theta \,|\, \mathbf{y}),$$

and subsequently approximating marginal state probabilities by their frequency in the sample

$$\mathbb{P}(q_t = s \,|\, \mathbf{y}) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(q_t^{(i)} = s).$$

Thus, for each position t, we get a complete probability distribution over the possible states. As the marginals of each variable are explicitly defined by conditioning on the other variables, an HMM lends itself to Gibbs sampling, i. e. repeatedly sampling from the marginals $(\mathbf{q} \,|\, \mathscr{A}, \theta, \mathbf{y})$, $(\mathscr{A} \,|\, \mathbf{q}, \theta, \mathbf{y})$ and $(\theta \,|\, \mathbf{q}, \mathscr{A}, \mathbf{y})$, conditioned on the previously sampled values. Using Bayes' formula and several conditional independence relations, the sampling process can be written as

$$\mathscr{A} \sim \mathbb{P}(\mathscr{A} \,|\, \mathbf{q}) \propto \mathbb{P}(\mathbf{q} \,|\, \mathscr{A}) \mathbb{P}(\mathscr{A} \,|\, \pi_{\mathscr{A}}),$$

$$\theta \sim \mathbb{P}(\theta \,|\, \mathbf{q}, \mathbf{y}) \propto \mathbb{P}(\mathbf{q}, \mathbf{y} \,|\, \theta) \mathbb{P}(\theta \,|\, \pi_\theta), \text{ and}$$

$$\mathbf{q} \sim \mathbb{P}(\mathbf{q} \,|\, \mathscr{A}, \mathbf{y}, \theta) \propto \mathbb{P}(\mathscr{A}, \mathbf{y}, \theta \,|\, \mathbf{q}) \mathbb{P}(\mathbf{q} \,|\, \pi_{\mathbf{q}}),$$

where π represents hyperparameters to the prior distribution. Notice that $\mathbf{q}$ depends on a prior only in q_0, so $\pi_{\mathbf{q}}$ is a set of parameters from which the distribution of q_0 is sampled (typically $\pi_{\mathbf{q}}$ are parameters of a Dirichlet distribution). In non-Bayesian settings, π often denotes $\mathbb{P}(q_0)$ itself. Typically, each prior will be conjugate, i. e. it will be the same class of distributions as the posterior. Thus π_S and $\pi_{A(k,:)}$, the hyperparameters of q_0 and the k-th row of $\mathscr{A}$, will be the α_i of a Dirichlet distribution, and $\pi_\theta = (\alpha, \beta, \nu, \mu_0)$ will be the parameters of a Normal-Inverse Gamma distribution.

Though there are several sampling schemes available, [58] has argued strongly in favor of Forward-Backward Gibbs sampling (FBG) [57]. Variations of this have been implemented for segmentation of aCGH data before [60, 62, 73]. However, sampling-based Bayesian methods such as FBG are computationally expensive. Recently, [62] have introduced *compressed FBG* by sampling over a shorter sequence of sufficient statistics of data segments which are likely to come from the same underlying state. Let $\mathbf{B} := (B_w)_{w=1}^{W}$ be a partition of $\mathbf{y}$ into W blocks. Each block B_w contains n_w elements. Let $y_{w,k}$ the k-th element in B_w.

The forward variable $\alpha_w(j)$ for this block needs to take into account the n_w emissions, the transitions into state j, and the $n_w - 1$ self-transitions, which yields

$$\alpha_w(j) := A_{jj}^{n_w-1}\mathbb{P}(B_w \mid \mu_j, \sigma_j^2) \sum_{i=1}^{n_w} \alpha_{w-1}(i) A_{ij}, \text{ and}$$

$$\mathbb{P}(B_w \mid \mu, \sigma^2) = \prod_{k=1}^{n_w} \mathbb{P}(y_{w,k} \mid \mu, \sigma^2).$$

The ideal block structure would correspond to the actual, unknown segmentation of the data. Any subdivision thereof would decrease the compression ratio, and thus the speedup, but still allow for recovery of the true breakpoints. In addition, such a segmentation would yield sufficient statistics for the likelihood computation that corresponds to the true parameters of the state generating a segment. Using wavelet theory, we show that such a block structure can be easily obtained.

Wavelet theory preliminaries

Here, we review some essential wavelet theory; for details, see [74, 75]. Let

$$\psi(x) := \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} \le x < 1 \\ 0 & \text{elsewhere} \end{cases}$$

be the Haar wavelet [76], and $\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k)$; j and k are called the *scale* and *shift* parameter. Any square-integrable function over the unit interval, $f \in L^2([0,1))$, can be approximated using the orthonormal basis $\{\psi_{j,k} \mid j,k \in \mathbb{Z}, -1 \le j, 0 \le k \le 2^j - 1\}$, admitting a *multiresolution analysis* [77, 78]. Essentially, this allows us to express a function $f(x)$ using scaled and shifted copies of one simple basis function $\psi(x)$ which is spatially localized, i. e. non-zero on only a finite interval in x. The Haar basis is particularly suited for expressing piecewise constant functions.

Finite data $\mathbf{y} := (y_0, \dots, y_{T-1})$ can be treated as an equidistant sample $f(\mathbf{x})$ by scaling the indices to the unit interval using $x_t := \frac{t}{T}$. Let $h := \log_2 T$. Then $\mathbf{y}$ can be expressed exactly as a linear combination over the Haar wavelet basis above, restricted to the maximum level of sampling resolution ($j \le h - 1$):

$$y_t = \sum_{j,k} d_{j,k}\psi_{j,k}(x_t).$$

The *wavelet transform* $\mathbf{d} = \mathscr{W}\mathbf{y}$ is an orthogonal endomorphism, and thus incurs neither redundancy nor loss of information. Surprisingly, $\mathbf{d}$ can be computed in linear time using the *pyramid algorithm* [77].

Compression via wavelet shrinkage

The Haar wavelet transform has an important property connecting it to block creation: Let $\hat{\mathbf{d}}$ be a vector obtained by setting elements of $\mathbf{d}$ to zero, then $\hat{\mathbf{y}} = \mathscr{W}^\intercal\hat{\mathbf{d}} := \hat{\mathscr{W}}^\intercal\mathbf{d}$ is called *selective wavelet reconstruction* (SW). If all coefficients $d_{j,k}$ with $\psi_{j,k}(x_t) \ne \psi_{j,k}(x_{t+1})$ are set to zero for some t, then $\hat{y}_t = \hat{y}_{t+1}$, which implies a block structure on $\hat{\mathbf{y}}$. Conversely, blocks of size > 2 (to account for some pathological cases) can only be created using SW. *This general equivalence between SW and compression is central to our method.* Note that $\hat{\mathbf{y}}$ does *not* have to be computed explicitly; the block boundaries can be inferred from the position of zero-entries in $\hat{\mathbf{d}}$ alone.

Assume all HMM states had the same emission variance σ^2. Since each state is associated with an emission mean, finding $\mathbf{q}$ can be viewed as a regression or smoothing problem of finding an estimate $\hat{\boldsymbol{\mu}}$ of a piecewise constant function $\boldsymbol{\mu}$ whose range is precisely the set of emission means, i. e.

$$\boldsymbol{\mu} = f(\mathbf{x}), \quad \mathbf{y} = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim_{\text{iid}} N(0, \sigma^2).$$

Unfortunately, regression methods typically do not limit the number of distinct values recovered, and will instead return some estimate $\hat{\mathbf{y}} \ne \hat{\boldsymbol{\mu}}$. However, if $\hat{\mathbf{y}}$ is piecewise constant and minimizes $\|\boldsymbol{\mu} - \hat{\mathbf{y}}\|$, the sample means of each block are close to the true emission means. This yields high likelihood for their corresponding state and hence a strong posterior distribution, leading to fast convergence. Furthermore, the change points in $\boldsymbol{\mu}$ must be close to change points in $\hat{\mathbf{y}}$, since moving block boundaries incurs additional loss, allowing for approximate recovery of true breakpoints. $\hat{\mathbf{y}}$ might however induce additional block boundaries that reduce the compression ratio.

In a series of ground-breaking papers, Donoho, Johnstone *et al.* [79–83] showed that SW could in theory be used as an almost ideal spatially adaptive regression method. Assuming one could provide an oracle $\Delta(\boldsymbol{\mu}, \mathbf{y})$ that would know the true $\boldsymbol{\mu}$, then there exists a method $M_{\text{SW}}(\mathbf{y}, \Delta) = \hat{\mathscr{W}}_{\text{SW}}^\intercal$ using an optimal subset of wavelet coefficients provided by Δ such that the quadratic risk of $\hat{\mathbf{y}}_{\text{SW}} := \hat{\mathscr{W}}_{\text{SW}}^\intercal\mathbf{d}$ is bounded as

$$\|\boldsymbol{\mu} - \hat{\mathbf{y}}_{\text{SW}}\|_2^2 = O\left(\frac{\sigma^2 \ln T}{T}\right).$$

By definition, M_{SW} would be the best compression method under the constraints of the Haar wavelet basis. This bound is generally unattainable, since the oracle cannot be queried. Instead, they have shown that for a method $M_{\text{WCT}}(\mathbf{y}, \lambda\sigma)$ called *wavelet coefficient thresholding*, which sets coefficients to zero whose absolute value is smaller than some threshold $\lambda\sigma$, there exists some $\lambda_T^\star \le \sqrt{2\ln T}$ with $\hat{\mathbf{y}}_{\text{WCT}} := \hat{\mathscr{W}}_{\text{WCT}}^\intercal\mathbf{d}$ such that

$$\|\boldsymbol{\mu} - \hat{\mathbf{y}}_{\text{WCT}}\|_2^2 \le (2\ln T + 1)\left(\|\boldsymbol{\mu} - \hat{\mathbf{y}}_{\text{SW}}\|_2^2 + \frac{\sigma^2}{T}\right).$$

This $\lambda_T^\star$ is minimax, i. e. the maximum risk incured over all possible data sets is not larger than that of any other threshold, and no better bound can be obtained. It is not easily computed, but for large T, on the order of tens to hundreds, the *universal threshold* $\lambda_T^{\text{u}} := \sqrt{2\ln T}$ is asymptotically minimax. In other words, for data large enough to warrant compression, universal thresholding is the best method to approximate $\boldsymbol{\mu}$, and thus the best wavelet-based compression scheme for a given noise level σ^2.

Integrating wavelet shrinkage into FBG

This compression method can easily be extended to multiple emission variances. Since we use a thresholding method, decreasing the variance simply subdivides existing blocks. If the threshold is set to the smallest emission variance among all states, $\hat{\mathbf{y}}$ will approximately preserve the breakpoints around those low-variance segments. Those of high variance are split into blocks of lower sample variance; see [84, 85] for an analytic expression. While the variances for the different states are not known, FBG provides *a priori* samples in each iteration. We hence propose the following simple adaptation: *In each sampling iteration, use the smallest sampled variance parameter to create a new block sequence via wavelet thresholding (Algorithm 1).*

Algorithm 1 Dynamically adaptive FBG for HMMs

procedure HAMMLET($\mathbf{y}$, $\pi_{\mathscr{A}}$, π_θ, $\pi_{\mathbf{q}}$)
    $T \leftarrow |\mathbf{y}|$
    $\lambda \leftarrow \sqrt{2 \ln T}$
    $\mathscr{A} \sim \mathbb{P}(\mathscr{A} \,|\, \pi_{\mathscr{A}})$
    $\theta \sim \mathbb{P}(\theta \,|\, \pi_\theta)$
    for $i = 1, \ldots, N$ **do**
        $\sigma_{\min} \leftarrow \min_{\sigma_i} \{\hat{\sigma}_{\mathrm{MAD}}, \sigma_i \,|\, \sigma_i^2 \in \theta\}$
        Create block sequence $\mathbf{B}$ from threshold $\lambda\sigma_{\min}$
        $\mathbf{q} \sim \mathbb{P}(\mathscr{A}, \mathbf{B}, \theta \,|\, \mathbf{q}) \mathbb{P}(\mathbf{q} \,|\, \pi_{\mathbf{q}})$
        $\mathscr{A} \sim \mathbb{P}(\mathbf{q} \,|\, \mathscr{A}) \mathbb{P}(\mathscr{A} \,|\, \pi_{\mathscr{A}})$
        $\theta \sim \mathbb{P}(\mathbf{q}, \mathbf{B} \,|\, \theta) \mathbb{P}(\theta \,|\, \pi_\theta)$
        Add count of marginal states for $\mathbf{q}$ to result
    end for
end procedure

While intuitively easy to understand, provable guarantees for the optimality of this method, specifically the correspondence between the wavelet and the HMM domain remain an open research topic. A potential problem could arise if all sampled variances are too large. In this case, blocks would be under-segmented, yield wrong posterior variances and hide possible state transitions. As a safeguard against over-compression, we use the standard method to estimate the variance of constant noise in shrinkage applications,

$$\hat{\sigma}^2_{\mathrm{MAD}} := \left(\frac{\mathrm{MAD}_k \{d_{\log_2 T - 1, k}\}}{\Phi^{-1}\left(\frac{3}{4}\right)} \right)^2$$

as an estimate of the variance in the dominant component, and modify the threshold definition to $\lambda \cdot \min\{\hat{\sigma}_{\mathrm{MAD}}, \sigma_i \in \theta\}$. If the data is not i.i.d., $\hat{\sigma}^2_{\mathrm{MAD}}$ will systematically underestimate the true variance [24]. In this case, the blocks get smaller than necessary, thus decreasing the compression.

A data structure for dynamic compression

The necessity to recreate a new block sequence in each iteration based on the most recent estimate of the smallest variance parameter creates the challenge of doing so with little computational overhead, specifically without repeatedly computing the inverse wavelet transform or considering all T elements in other ways. We achieve this by creating a simple tree-based data structure.

The pyramid algorithm yields $\mathbf{d}$ sorted according to (j,k). Again, let $h := \log_2 T$, and $\ell := h - j$. We can map the wavelet $\psi_{j,k}$ to a perfect binary tree of height h such that all wavelets for scale j are nodes on level ℓ, nodes within each level are sorted according to k, and ℓ is increasing from the leaves to the root (Fig. 7). $\mathbf{d}$ represents a breadth-first search (BFS) traversal of that tree, with $d_{j,k}$ being the entry at position $\lfloor 2^j \rfloor + k$. Adding y_i as the i-th leaf on level $\ell = 0$, each non-leaf node represents a wavelet which is non-zero for the $n := 2^\ell$ data points y_t, for t in the the interval $I_{j,k} := [kn, (k+1)n - 1]$ stored in the leaves below; notice that for the leaves, $kn = t$.

This implies that the leaves in any subtree all have the same value after wavelet thresholding if all the wavelets in this subtree are set to zero. We can hence avoid computing the inverse wavelet transform to create blocks. Instead, each node stores the maximum absolute wavelet coefficient in the entire subtree, as well as the sufficient statistics required for calculating the likelihood function. More formally, a node $N_{\ell,t}$ corresponds to wavelet $\psi_{j,k}$, with $\ell = h - j$ and $t = k2^\ell$ ($\psi_{-1,0}$ is simply constant on the $[0,1)$ interval and has no effect on block creation, thus we discard it). Essentially, ℓ numbers the levels beginning at the leaves, and t marks the start position of the block when pruning the subtree rooted at $N_{\ell,t}$. The members stored in each node are:

- The number of leaves, corresponding to the block size:

$$N_{\ell,t}[n] := 2^\ell$$

- The sum of data points stored in the subtree leaves:

$$N_{\ell,t}[\Sigma_1] := \sum_{i \in I_{j,k}} y_i$$

- Similarly, the sum of squares:

$$N_{\ell,t}[\Sigma_2] := \sum_{i \in I_{j,k}} y_i^2$$

- The maximum absolute wavelet coefficient of the subtree, including the current $d_{j,k}$ itself:

$$N_{0,t}[d] := 0 \qquad N_{\ell>0,t}[d] := \max_{\substack{\ell' \le \ell \\ t \le t' < t + 2^\ell}} \left\{ \left| d_{h-\ell', 2^{\ell'}/t'} \right| \right\}$$

All these values can be computed recursively from the child nodes in linear time. As some real data sets contain salt-and-pepper noise, which manifests as isolated large coefficients on the lowest level, its is possible to ignore the first level in the maximum computation so that no information to create a single-element block for outliers is passed up the tree. We refer to this technique as *noise control*. Notice that this does not imply that blocks are only created at even t, since true transitions manifest in coefficients on multiple levels.

The block creation algorithm is simple: upon construction, the tree is converted to *depth-first search* (DFS) order, which simply amounts to sorting the BFS array according to (kn, j). Given a threshold, the tree is then traversed in DFS order by iterating linearly over the array (Fig. 7, solid lines). Once the maximum coefficient stored in a node is less or equal to the threshold, a block of size n is created, and the entire subtree is skipped (dashed lines). As the tree is perfect binary and complete, the next array position in DFS traversal after pruning the subtree rooted at the node at index i is simply obtained as $i + 2n - 1$, so no expensive pointer structure needs to be maintained, leaving the tree data structure a simple flat array. An example of dynamic block creation is given in Fig. 8.

Once the Gibbs sampler converges to a set of variances, the block structure is less likely to change. To avoid recreating the same block structure over and over again, we employ a technique called *block structure prediction*. Since the different block structures are subdivisions of each other that occur in a specific order for decreasing σ^2, there is a simple bijection between the number of blocks and the block structure itself. Thus, for each block sequence length we register the minimum and maximum variance that creates that sequence. Upon entering a new iteration, we check if the current variance would create the same number of blocks as in the previous iteration, which guarantees that we would obtain the same block sequence, and hence can avoid recomputation.

The wavelet tree data structure can be readily extended to multivariate data of dimensionality m. Instead of storing m different trees and reconciling m different block patterns in each iteration, one simply stores m different values for each sufficient statistic in a tree node. Since we have to traverse into the combined tree if the coefficient of any of the m trees was below the threshold, we simply store the largest $N_{\ell,t}[d]$ among the corresponding nodes of the trees, which means that the block creation can be done in $O(T)$ instead of $O(mT)$, i. e. the dimensionality of the data only enters into the creation of the data structure, but not the query during sampling iterations.

Automatic priors

While Bayesian methods allow for inductive bias such as the expected location of means, it is desirable to be able to use our method even when little domain knowledge exists, or large variation is expected, such as the lab and batch effects commonly observed in micro-arrays [86], as well as unknown means due to sample contamination. Since FBG does require a prior even in that case, we propose the

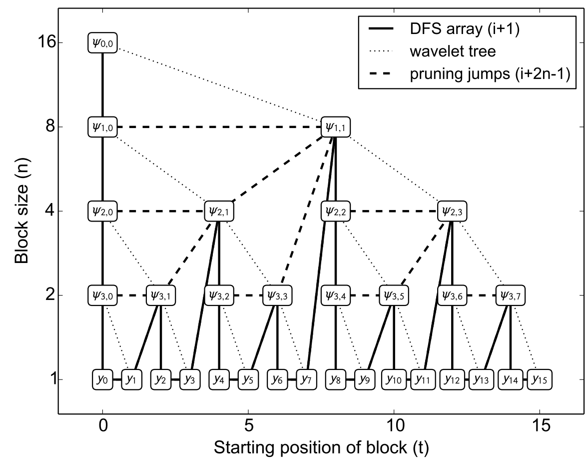

Figure 7. Mapping of wavelets $\psi_{j,k}$ and data points y_t to tree nodes $N_{\ell,t}$. Each node is the root of a subtree with $n = 2^\ell$ leaves; pruning that subtree yields a block of size n, starting at position t. For instance, the node $N_{1,6}$ is located at position 13 of the DFS array (solid line), and corresponds to the wavelet $\psi_{3,3}$. A block of size $n = 2$ can be created by pruning the subtree, which amounts to advancing by $2n - 1 = 3$ positions (dashed line), yielding $N_{3,8}$ at position 16, which is the wavelet $\psi_{1,1}$. Thus the number of steps for creating blocks per iteration is at most the number of nodes in the tree, and thus strictly smaller than $2T$.

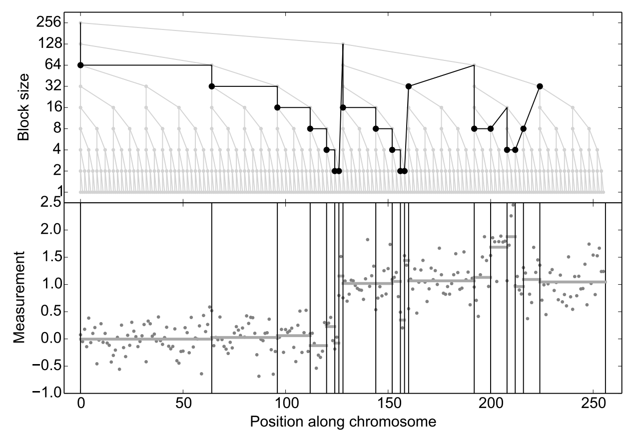

Figure 8. Example of dynamic block creation. The data is of size $T = 256$, so the wavelet tree contains 512 nodes. Here, only 37 entries had to be checked against the threshold (dark line), 19 of which (round markers) yielded a block (vertical lines on the bottom). Sampling is hence done on a short array of 19 blocks instead of 256 individual values, thus the compression ratio is 13.5. The horizontal lines in the bottom subplot are the block means derived from the sufficient statistics in the nodes. Notice how the algorithm creates small blocks around the breakpoints, e. g. at $t \approx 125$, which requires traversing to lower levels and thus induces some additional blocks in other parts of the tree (left subtree), since all block sizes are powers of 2. This somewhat reduces the compression ratio, which is unproblematic as it increases the degrees of freedom in the sampler.

following method to specify hyperparameters of a weak prior automatically. Posterior samples of means and variances are drawn from a Normal-Inverse Gamma distribution $(\mu, \sigma^2) \sim \mathrm{NI}\Gamma(\mu_0, \nu, \alpha, \beta)$, whose marginals simply separate into a Normal and an Inverse Gamma distribution

$$\sigma^2 \sim \mathrm{I}\Gamma(\alpha, \beta), \quad \mu \sim N\left(\mu_0, \frac{\sigma^2}{\nu}\right).$$

Let s^2 be a user-defined variance (or automatically infered, e. g. from the largest of the finest detail coefficients, or $\hat{\sigma}^2_{\mathrm{MAD}}$), and p the desired probability to sample a variance not larger than s^2. From the CDF of $\mathrm{I}\Gamma$ we obtain

$$p := \mathbb{P}(\sigma^2 \leq s^2) = \frac{\Gamma\left(\alpha, \frac{\beta}{s^2}\right)}{\Gamma(\alpha)} = Q\left(\alpha, \frac{\beta}{s^2}\right).$$

$\mathrm{I}\Gamma$ has a mean for $\alpha > 1$, and closed-form solutions for $\alpha \in \mathbb{N}$. Furthermore, $\mathrm{I}\Gamma$ has positive skewness for $\alpha > 3$. We thus let $\alpha = 2$, which yields

$$\beta = -s^2\left(W_{-1}\left(-\frac{p}{e}\right) + 1\right), \quad 0 < p \leq 1,$$

where W_{-1} is the negative branch of the Lambert W-function, which is transcendental. However, an excellent analytical approximation with a maximum error of 0.025% is given in [87], which yields

$$\beta \approx s^2\left(\frac{2\sqrt{b}}{M_1\sqrt{b} + \sqrt{2}\left(M_2 b \exp\left(M_3\sqrt{b}\right) + 1\right)} + b\right),$$

$$b := -\ln p,$$

$$M_1 := 0.3361, \quad M_2 := -0.0042, \quad M_3 := -0.0201.$$

Since the mean of $\mathrm{I}\Gamma$ is $\frac{\beta}{\alpha-1}$, the expected variance of μ is $\frac{\beta}{\nu}$ for $\alpha = 2$. To ensure proper mixing, we could simply set $\frac{\beta}{\mu}$ to the sample variance of the data, which can be estimated from the sufficient statistics in the root of the wavelet tree (the first entry in the array), provided that $\boldsymbol{\mu}$ contained all states in almost equal number. However, due to possible class imbalance, means for short segments far away from μ_0 can have low sampling probability, as they do not contribute much to the sample variance of the data. We thus define δ to be the sample variance of block means in the compression obtained by $\hat{\sigma}^2_{\mathrm{MAD}}$, and take the maximum of those two variances. We thus obtain

$$\mu_0 := \frac{\Sigma_1}{n}, \text{ and } \nu = \beta \max\left\{\frac{n\Sigma_2 - \Sigma_1^2}{n^2}, \delta\right\}^{-1}.$$

Numerical issues

To assure numerical stability, many HMM implementations resort to log-space computations. Our implementation, which differs from [62], reflects the block structure and avoids a considerable number of expensive function calls (exp, log, pow) required by others. This yields an additional speedup of about one order of magnitude compared to the log version (data not shown). The term accounting for emissions and self-transitions within the block can be written as

$$\frac{A_{jj}^{n_w-1}}{(2\pi)^{n_w/2}\sigma_j^{n_w}} \exp\left(-\sum_{k=1}^{n_w} \frac{(y_{w,k} - \mu_j)^2}{2\sigma_j^2}\right).$$

Any constant cancels out during normalization. Furthermore, exponentiation of potentially small numbers causes underflows. We hence move those terms into the exponent, utilizing the much stabler logarithm function.

$$\exp\left(-\sum_{k=1}^{n_w} \frac{(y_{w,k} - \mu_j)^2}{2\sigma_j^2} + (n_w - 1)\log A_{jj} - n_w \log \sigma_j\right).$$

Using the block's sufficient statistics

$$n_w, \quad \Sigma_1 := \sum_{k=1}^{n_w} y_{w,k}, \quad \Sigma_2 := \sum_{k=1}^{n_w} y_{w,k}^2.$$

the exponent can be rewritten as

$$E_w(j) := \frac{2\mu_j\Sigma_1 - \Sigma_2}{2\sigma_j^2} + K(n_w, j),$$

$$K(n_w, j) := (n_w - 1)\log A_{jj} - n_w\left(\log \sigma_j + \frac{\mu_j^2}{2\sigma_j^2}\right).$$

Note that $K(n_w, j)$ can be precomputed for each iteration, thus greatly reducing the number of expensive function calls. To avoid overflow of the exponential function, we subtract the largest such exponents among all states, hence $E_w(j) \leq 0$. This is equivalent to dividing the forward variables by

$$\exp\left(\max_k E_w(k)\right),$$

which cancels out during normalization. Hence we obtain

$$\tilde{\alpha}_w(j) := \exp\left(E_w(j) - \max_k E_w(k)\right)\sum_{i=1}^{n_w} \alpha_{w-1}(i)A_{ij},$$

which are then normalized to

$$\hat{\alpha}_w(j) = \frac{\tilde{\alpha}_w(j)}{\sum_k \tilde{\alpha}_w(k)}.$$

Availability of supporting data

The implementation as well as the supplemental data are available at `http://bioinformatics.rutgers.edu/Software/HaMMLET/`.

Competing interests

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Authors' contributions

JW and AS conceived and supervised the study, and wrote the paper. EB and JW developed the implementation. All authors discussed, read and approved the article.

Acknowledgements

We would like to thank Md Pavel Mahmud for helpful advice. E. Brugel participated in the DIMACS REU program with support from NSF award 1263082.

References

[1] Iafrate, A.J., Feuk, L., Rivera, M.N., Listewnik, M.L., Donahoe, P.K., Qi, Y., Scherer, S.W., Lee, C.: Detection of large-scale variation in the human genome. Nature Genetics **36**(9), 949–51 (2004). doi:10.1038/ng1416

[2] Feuk, L., Marshall, C.R., Wintle, R.F., Scherer, S.W.: Structural variants: changing the landscape of chromosomes and design of disease studies. Human Molecular Genetics **15 Spec No**, 57–66 (2006). doi:10.1093/hmg/ddl057

[3] McCarroll, S.A., Altshuler, D.M.: Copy-number variation and association studies of human disease. Nature Genetics **39**(7 Suppl), 37–42 (2007). doi:10.1038/ng2080

[4] Wain, L.V., Armour, J.A.L., Tobin, M.D.: Genomic copy number variation, human health, and disease. Lancet **374**(9686), 340–350 (2009). doi:10.1016/S0140-6736(09)60249-X

[5] Beckmann, J.S., Sharp, A.J., Antonarakis, S.E.: CNVs and genetic medicine (excitement and consequences of a rediscovery). Cytogenetic and Genome Research **123**(1-4), 7–16 (2008). doi:10.1159/000184687

[6] Cook, E.H., Scherer, S.W.: Copy-number variations associated with neuropsychiatric conditions. Nature **455**(7215), 919–23 (2008). doi:10.1038/nature07458

[7] Buretić-Tomljanović, A., Tomljanović, D.: Human genome variation in health and in neuropsychiatric disorders. Psychiatria Danubina **21**(4), 562–9 (2009)

[8] Merikangas, A.K., Corvin, A.P., Gallagher, L.: Copy-number variants in neurodevelopmental disorders: promises and challenges. Trends in Genetics **25**(12), 536–44 (2009). doi:10.1016/j.tig.2009.10.006

[9] Cho, E.K., Tchinda, J., Freeman, J.L., Chung, Y.-J., Cai, W.W., Lee, C.: Array-based comparative genomic hybridization and copy number variation in cancer research. Cytogenetic and Genome Research **115**(3-4), 262–72 (2006). doi:10.1159/000095923

[10] Shlien, A., Malkin, D.: Copy number variations and cancer susceptibility. Current Opinion in Oncology **22**(1), 55–63 (2010). doi:10.1097/CCO.0b013e328333dca4

[11] Feuk, L., Carson, A.R., Scherer, S.W.: Structural variation in the human genome. Nature Reviews Genetics **7**(2), 85–97 (2006). doi:10.1038/nrg1767

[12] Beckmann, J.S., Estivill, X., Antonarakis, S.E.: Copy number variants and genetic traits: closer to the resolution of phenotypic to genotypic variability. Nature Reviews Genetics **8**(8), 639–46 (2007). doi:10.1038/nrg2149

[13] Sharp, A.J.: Emerging themes and new challenges in defining the role of structural variation in human disease. Human Mutation **30**(2), 135–44 (2009). doi:10.1002/humu.20843

[14] Garnis, C., Coe, B.P., Zhang, L., Rosin, M.P., Lam, W.L.: Overexpression of LRP12, a gene contained within an 8q22 amplicon identified by high-resolution array CGH analysis of oral squamous cell carcinomas. Oncogene **23**(14), 2582–6 (2004). doi:10.1038/sj.onc.1207367

[15] Albertson, D.G., Ylstra, B., Segraves, R., Collins, C., Dairkee, S.H., Kowbel, D., Kuo, W.L., Gray, J.W., Pinkel, D.: Quantitative mapping of amplicon structure by array CGH identifies CYP24 as a candidate oncogene. Nature Genetics **25**(2), 144–6 (2000). doi:10.1038/75985

[16] Veltman, J.A., Fridlyand, J., Pejavar, S., Olshen, A.B., Korkola, J.E., DeVries, S., Carroll, P., Kuo, W.-L., Pinkel, D., Albertson, D.G., Cordon-Cardo, C., Jain, A.N., Waldman, F.M.: Array-based comparative genomic hybridization for genome-wide screening of DNA copy number in bladder tumors. Cancer Research **63**(11), 2872–80 (2003)

[17] Autio, R., Hautaniemi, S., Kauraniemi, P., Yli-Harja, O., Astola, J., Wolf, M., Kallioniemi, A.: CGH-Plotter: MATLAB toolbox for CGH-data analysis. Bioinformatics **19**(13), 1714–1715 (2003). doi:10.1093/bioinformatics/btg230

[18] Pollack, J.R., Sørlie, T., Perou, C.M., Rees, C.A., Jeffrey, S.S., Lonning, P.E., Tibshirani, R., Botstein, D., Børresen-Dale, A.-L., Brown, P.O.: Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. Proceedings of the National Academy of Sciences of the United States of America **99**(20), 12963–8 (2002). doi:10.1073/pnas.162471999

[19] Eilers, P.H.C., de Menezes, R.X.: Quantile smoothing of array CGH data. Bioinformatics **21**(7), 1146–53 (2005). doi:10.1093/bioinformatics/bti148

[20] Cleveland, W.S.: Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association **74**(368) (1979). doi:10.1080/01621459.1979.10481038

[21] Beheshti, B., Braude, I., Marrano, P., Thorner, P., Zielenska, M., Squire, J.A.: Chromosomal localization of DNA amplifications in neuroblastoma tumors using cDNA microarray comparative genomic hybridization. Neoplasia **5**(1), 53–62 (2003)

[22] Polzehl, J., Spokoiny, V.G.: Adaptive weights smoothing with applications to image restoration. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **62**(2), 335–354 (2000). doi:10.1111/1467-9868.00235

[23] Hupé, P., Stransky, N., Thiery, J.-P., Radvanyi, F., Barillot, E.: Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. Bioinformatics **20**(18), 3413–22 (2004). doi:10.1093/bioinformatics/bth418

[24] Wang, Y., Wang, S.: A novel stationary wavelet denoising algorithm for array-based DNA Copy Number data. International Journal of Bioinformatics Research and Applications **3**(x), 206–222 (2007). doi:10.1504/IJBRA.2007.013603

[25] Hsu, L., Self, S.G., Grove, D., Randolph, T., Wang, K., Delrow, J.J., Loo, L., Porter, P.: Denoising array-based comparative genomic hybridization data using wavelets. Biostatistics (Oxford, England) **6**(2), 211–26 (2005). doi:10.1093/biostatistics/kxi004

[26] Nguyen, N., Huang, H., Oraintara, S., Vo, A.: A New Smoothing Model for Analyzing Array CGH Data. In: Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, Boston, MA (2007). doi:10.1109/BIBE.2007.4375683. `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4375683`

[27] Nguyen, N., Huang, H., Oraintara, S., Vo, A.: Stationary wavelet packet transform and dependent laplacian bivariate shrinkage estimator for array-CGH data smoothing. Journal of Computational Biology **17**(2), 139–52 (2010). doi:10.1089/cmb.2009.0013

[28] Huang, H., Nguyen, N., Oraintara, S., Vo, A.: Array CGH data modeling and smoothing in Stationary Wavelet Packet Transform domain. BMC Genomics **9 Suppl 2**, 17 (2008). doi:10.1186/1471-2164-9-S2-S17

[29] Holt, C., Losic, B., Pai, D., Zhao, Z., Trinh, Q., Syam, S., Arshadi, N., Jang, G.H., Ali, J., Beck, T., McPherson, J., Muthuswamy, L.B.: WaveCNV: allele-specific copy number alterations in primary tumors and xenograft models from next-generation sequencing. Bioinformatics, 611 (2013). doi:10.1093/bioinformatics/btt611

[30] Price, T.S., Regan, R., Mott, R., Hedman, A., Honey, B., Daniels, R.J., Smith, L., Greenfield, A., Tiganescu, A., Buckle, V., Ventress, N., Ayyub, H., Salhan, A., Pedraza-Diaz, S., Broxholme, J., Ragoussis, J., Higgs, D.R., Flint, J., Knight, S.J.L.: SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. Nucleic Acids Research **33**(11), 3455–3464 (2005). doi:10.1093/nar/gki643

[31] Tsourakakis, C.E., Peng, R., Tsiarli, M.A., Miller, G.L., Schwartz, R.: Approximation algorithms for speeding up dynamic programming and denoising aCGH data. Journal of Experimental Algorithmics **16**, 1–1 (2011). doi:10.1145/1963190.2063517

[32] Olshen, A.B., Venkatraman, E.S.: Change-point analysis of array-based comparative genomic hybridization data. ASA Proceedings of the Joint Statistical Meetings, 2530–2535 (2002)

[33] Olshen, A.B., Venkatraman, E.S., Lucito, R., Wigler, M.: Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics (Oxford, England) **5**(4), 557–572 (2004). doi:10.1093/biostatistics/kxh008

[34] Venkatraman, E.S., Olshen, A.B.: A faster circular binary segmentation algorithm for the analysis of array CGH data. Bioinformatics **23**(6), 657–63 (2007). doi:10.1093/bioinformatics/btl646

[35] Picard, F., Robin, S., Lavielle, M., Vaisse, C., Daudin, J.-J.: A statistical approach for array CGH data analysis. BMC Bioinformatics **6**(1), 27 (2005). doi:10.1186/1471-2105-6-27

[36] Myers, C.L., Dunham, M.J., Kung, S.Y., Troyanskaya, O.G.: Accurate detection of aneuploidies in array CGH and gene expression microarray data. Bioinformatics **20**(18), 3533–43 (2004). doi:10.1093/bioinformatics/bth440

[37] Wang, P., Kim, Y., Pollack, J., Narasimhan, B., Tibshirani, R.: A method for calling gains and losses in array CGH data. Biostatistics (Oxford, England) **6**(1), 45–58 (2005). doi:10.1093/biostatistics/kxh017

[38] Xing, B., Greenwood, C.M.T., Bull, S.B.: A hierarchical clustering method for estimating copy number variation. Biostatistics (Oxford, England) **8**(3), 632–53 (2007). doi:10.1093/biostatistics/kxl035

[39] Chen, C.-H., Lee, H.-C.C., Ling, Q., Chen, H.-R., Ko, Y.-A., Tsou, T.-S., Wang, S.-C., Wu, L.-C., Lee, H.-C.C.: An all-statistics, high-speed algorithm for the analysis of copy number variation in genomes. Nucleic Acids Research **39**(13), 89 (2011). doi:10.1093/nar/gkr137

[40] Jong, K., Marchiori, E., van der Vaart, A., Ylstra, B., Weiss, M., Meijer, G.: Chromosomal Breakpoint Detection in Human Cancer. Lecture Notes in Computer Science, vol. 2611. Springer, Berlin, Heidelberg (2003). doi:10.1007/3-540-36605-9. http://link.springer.com/10.1007/3-540-36605-9

[41] Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains (1966). doi:10.1214/aoms/1177699147. http://www.jstor.org/stable/2238772?seq=1#page_scan_tab_contentshttp://www.jstor.org/stable/2238772 Accessed 2015-02-14

[42] Snijders, A.M., Fridlyand, J., Mans, D.A., Segraves, R., Jain, A.N., Pinkel, D., Albertson, D.G.: Shaping of tumor and drug-resistant genomes by instability and selection. Oncogene **22**(28), 4370–9 (2003). doi:10.1038/sj.onc.1206482

[43] Sebat, J., Lakshmi, B., Troge, J., Alexander, J., Young, J., Lundin, P., Må nér, S., Massa, H., Walker, M., Chi, M., Navin, N., Lucito, R., Healy, J., Hicks, J., Ye, K., Reiner, A., Gilliam, T.C., Trask, B., Patterson, N., Zetterberg, A., Wigler, M.: Large-scale copy number polymorphism in the human genome. Science **305**(5683), 525–8 (2004). doi:10.1126/science.1098918

[44] Sebat, J., Lakshmi, B., Malhotra, D., Troge, J., Lese-Martin, C., Walsh, T., Yamrom, B., Yoon, S., Krasnitz, A., Kendall, J., Leotta, A., Pai, D., Zhang, R., Lee, Y.-H., Hicks, J., Spence, S.J., Lee, A.T., Puura, K., Lehtimäki, T., Ledbetter, D., Gregersen, P.K., Bregman, J., Sutcliffe, J.S., Jobanputra, V., Chung, W., Warburton, D., King, M.-C., Skuse, D., Geschwind, D.H., Gilliam, T.C., Ye, K., Wigler, M.: Strong association of de novo copy number mutations with autism. Science **316**(5823), 445–449 (2007). doi:10.1126/science.1138659

[45] Fridlyand, J., Snijders, A.M., Pinkel, D., Albertson, D.G., Jain, A.N.: Hidden Markov models approach to the analysis of array CGH data. Journal of Multivariate Analysis **90**(1), 132–153 (2004). doi:10.1016/j.jmva.2004.02.008

[46] Zhao, X.: An Integrated View of Copy Number and Allelic Alterations in the Cancer Genome Using Single Nucleotide Polymorphism Arrays. Cancer Research **64**(9), 3060–3071 (2004). doi:10.1158/0008-5472.CAN-03-3308

[47] de Vries, B.B.A., Pfundt, R., Leisink, M., Koolen, D.A., Vissers, L.E.L.M., Janssen, I.M., van Reijmersdal, S., Nillesen, W.M., Huys, E.H.L.P.G., de Leeuw, N., Smeets, D., Sistermans, E.A., Feuth, T., van Ravenswaaij-Arts, C.M.A., van Kessel, A.G., Schoenmakers, E.F.P.M., Brunner, H.G., Veltman, J.A., van Reijmersdal, S., Nillesen, W.M., Huys, E.H.L.P.G., de Leeuw, N., Smeets, D., Sistermans, E.A., Feuth, T., van Ravenswaaij-Arts, C.M.A., van Kessel, A.G., Schoenmakers, E.F.P.M., Brunner, H.G., Veltman, J.A.: Diagnostic genome profiling in mental retardation. American Journal of Human Genetics **77**(4), 606–616 (2005). doi:10.1086/491719

[48] Nannya, Y., Sanada, M., Nakazaki, K., Hosoya, N., Wang, L., Hangaishi, A., Kurokawa, M., Chiba, S., Bailey, D.K., Kennedy, G.C., Ogawa, S.: A robust algorithm for copy number detection using high-density oligonucleotide single nucleotide polymorphism genotyping arrays. Cancer Research **65**(14), 6071–9 (2005). doi:10.1158/0008-5472.CAN-05-0465

[49] Marioni, J.C., Thorne, N.P., Tavaré, S.: BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data. Bioinformatics **22**(9), 1144–6 (2006). doi:10.1093/bioinformatics/btl089

[50] Korbel, J.O., Urban, A.E., Grubert, F., Du, J., Royce, T.E., Starr, P., Zhong, G., Emanuel, B.S., Weissman, S.M., Snyder, M., Gerstein, M.B.: Systematic prediction and validation of breakpoints associated with copy-number variants in the human genome. Proceedings of the National Academy of Sciences of the United States of America **104**(24), 10110–10115 (2007). doi:10.1073/pnas.0703834104

[51] Cahan, P., Godfrey, L.E., Eis, P.S., Richmond, T.A., Selzer, R.R., Brent, M., McLeod, H.L., Ley, T.J., Graubert, T.A.: wuHMM: a robust algorithm to detect DNA copy number variation using long oligonucleotide microarray data. Nucleic Acids Research **36**(7), 41 (2008). doi:10.1093/nar/gkn110

[52] Rueda, O.M., Diaz-Uriarte, R.: RJaCGH: Bayesian analysis of aCGH arrays for detecting copy number changes and recurrent regions. Bioinformatics **25**(15), 1959–1960 (2009). doi:10.1093/bioinformatics/btp307

[53] Bilmes, J.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models (1998). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.613

[54] Rabiner, L.R.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE **77**, 257–286 (1989). doi:10.1109/5.18626

[55] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory **13**(2), 260–269 (1967). doi:10.1109/TIT.1967.1054010

[56] Forney, G.D.: The Viterbi algorithm. Proceedings of the IEEE **61**(3), 268–278 (1973). doi:10.1109/PROC.1973.9030

[57] Chib, S.: Calculating posterior distributions and modal estimates in Markov mixture models. Journal of Econometrics **75**(1), 79–97 (1996)

[58] Scott, S.L.: Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. Journal of the American Statistical Association **97**(457), 337–351 (2002)

[59] Guha, S., Li, Y., Neuberg, D.: Bayesian Hidden Markov Modeling of Array CGH Data (2006). http://biostats.bepress.com/harvardbiostat/paper24

[60] Shah, S.P., Xuan, X., DeLeeuw, R.J., Khojasteh, M., Lam, W.L., Ng, R., Murphy, K.P.: Integrating copy number polymorphisms into array CGH analysis using a robust HMM. Bioinformatics **22**(14), 431–9 (2006). doi:10.1093/bioinformatics/btl238

[61] Shah, S.P., Lam, W.L., Ng, R.T., Murphy, K.P.: Modeling recurrent DNA copy number alterations in array CGH data. Bioinformatics **23**(13), 450–8 (2007). doi:10.1093/bioinformatics/btm221

[62] Mahmud, M.P., Schliep, A.: Fast MCMC sampling for Hidden Markov Models to determine copy number variations. BMC Bioinformatics **12**, 428 (2011). doi:10.1186/1471-2105-12-428

[63] Ben-Yaacov, E., Eldar, Y.C.: A fast and flexible method for the segmentation of aCGH data. Bioinformatics **24**(16), 139–45 (2008). doi:10.1093/bioinformatics/btn272

[64] Wang, J., Meza-Zepeda, L.A., Kresse, S.H., Myklebost, O.: M-CGH: analysing microarray-based CGH experiments. BMC Bioinformatics **5**(1), 74 (2004). doi:10.1186/1471-2105-5-74

[65] Lai, W.R., Johnson, M.D., Kucherlapati, R., Park, P.J.: Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. Bioinformatics **21**(19), 3763–3770 (2005). doi:10.1093/bioinformatics/bti611

[66] Willenbrock, H., Fridlyand, J.: A comparison study: applying segmentation to array CGH data for downstream analyses. Bioinformatics **21**(22), 4084–91 (2005). doi:10.1093/bioinformatics/bti677

[67] Hodgson, G., Hager, J.H., Volik, S., Hariono, S., Wernick, M., Moore, D., Nowak, N., Albertson, D.G., Pinkel, D., Collins, C., Hanahan, D., Gray, J.W.: Genome scanning with array CGH delineates regional alterations in mouse islet carcinomas. Nature Genetics **29**(4), 459–64 (2001). doi:10.1038/ng771

[68] Edgren, H., Murumagi, A., Kangaspeska, S., Nicorici, D., Hongisto, V., Kleivi, K., Rye, I.H., Nyberg, S., Wolf, M., Børresen-Dale, A.-L., Kallioniemi, O.: Identification of fusion genes in breast cancer by paired-end RNA-sequencing. Genome Biology **12**(1), 6 (2011). doi:10.1186/gb-2011-12-1-r6

[69] Burdall, S., Hanby, A., Lansdown, M., Speirs, V.: Breast cancer cell lines: friend or foe? Breast Cancer Research **5**(2), 89–95 (2003). doi:10.1186/bcr577

[70] Holliday, D.L., Speirs, V.: Choosing the right cell line for breast cancer research. Breast Cancer Research **13**(4), 215 (2011). doi:10.1186/bcr2889

[71] Özgür, A., Özgür, L., Güngör, T.: Text Categorization with Class-Based and Corpus-Based Keyword Selection. Proceedings of the 20th International Conference on Computer and Information Sciences **3733**, 606–615 (2005). doi:10.1007/11569596

[72] Snijders, A.M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., 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.: Assembly of microarrays for genome-wide measurement of DNA copy number. Nature Genetics **29**(3), 263–4 (2001). doi:10.1038/ng754

[73] Mahmud, M.P., Schliep, A.: Speeding up Bayesian HMM by the four Russians method. In: Proceedings of the 11th International Conference on Algorithms in Bioinformatics, pp. 188–200 (2011). http://dl.acm.org/citation.cfm?id=2039945.2039962

[74] Daubechies, I.: Ten Lectures on Wavelets vol. 61, p. 357 (1992). doi:10.1137/1.9781611970104. http://epubs.siam.org/doi/book/10.1137/1.9781611970104

[75] Mallat, S.G.: A Wavelet Tour of Signal Processing: The Sparse Way. Academic Press, Burlington, MA (2009). http://dl.acm.org/citation.cfm?id=1525499

[76] Haar, A.: Zur Theorie der orthogonalen Funktionensysteme. Mathematische Annalen **69**(3), 331–371 (1910). doi:10.1007/BF01456326

[77] Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(7), 674–693 (1989). doi:10.1109/34.192463

[78] Mallat, S.G.: Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. Transactions of the American Mathematical Society **315**(1), 69–69 (1989). doi:10.1090/S0002-9947-1989-1008470-5

[79] Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**(3), 425–455 (1994). doi:10.1093/biomet/81.3.425

[80] Donoho, D.L., Johnstone, I.M.: Asymptotic minimaxity of wavelet estimators with sampled data. Statistica Sinica **9**, 1–32 (1999)

[81] Donoho, D.L., Johnstone, I.M.: Minimax estimation via wavelet shrinkage. The Annals of Statistics **26**(3), 879–921 (1998)

[82] Donoho, D.L., Johnstone, I.M.: Threshold selection for wavelet shrinkage of noisy data. In: Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 24–25. IEEE, Baltimore, MD (1994). doi:10.1109/IEMBS.1994.412133. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=412133

[83] Donoho, D.L., Johnstone, I.M., Kerkyacharian, G., Picard, D.: Wavelet Shrinkage: Asymptopia? Journal of the Royal Statistical Society: Series B (Statistical Methodology) **57**(2), 301–369 (1995)

[84] Percival, D.B., Mofjeld, H.O.: Analysis of Subtidal Coastal Sea Level Fluctuations Using Wavelets. Journal of the American Statistical Association **92**(439), 868 (1997). doi:10.2307/2965551

[85] Serroukh, A., Walden, A.T., Percival, D.B.: Statistical Properties and Uses of the Wavelet Variance Estimator for the Scale Analysis of Time Series (2012). doi:10.1080/01621459.2000.10473913

[86] Luo, J., Schumacher, M., Scherer, A., Sanoudou, D., Megherbi, D., Davison, T., Shi, T., Tong, W., Shi, L., Hong, H., Zhao, C., Elloumi, F., Shi, W., Thomas, R., Lin, S., Tillinghast, G., Liu, G., Zhou, Y., Herman, D., Li, Y., Deng, Y., Fang, H., Bushel, P., Woods, M., Zhang, J.: A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. The Pharmacogenomics Journal **10**(4), 278–91 (2010). doi:10.1038/tpj.2010.57

[87] Barry, D.A., Parlange, J.-Y., Li, L., Prommer, H., Cunningham, C.J., Stagnitti, F.: Analytical approximations for real values of the Lambert W-function. Mathematics and Computers in Simulation **53**(1-2), 95–103 (2000). doi:10.1016/S0378-4754(00)00172-5