BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 7 2013, pages 878–885
doi:10.1093/bioinformatics/btt065

Genetics and population analysis

Advance Access publication February 13, 2013

A dynamic Bayesian Markov model for phasing and characterizing haplotypes in next-generation sequencing

Yu Zhang

Department of Statistics, The Pennsylvania State University, 325 Thomas, University Park, PA 16802, USA

Associate Editor: Jeffrey Barrett

ABSTRACT

Motivation: Next-generation sequencing (NGS) technologies have enabled whole-genome discovery and analysis of genetic variants in many species of interest. Individuals are often sequenced at low coverage for detecting novel variants, phasing haplotypes and inferring population structures. Although several tools have been developed for SNP and genotype calling in NGS data, haplotype phasing is often done separately on the called genotypes.

Results: We propose a dynamic Bayesian Markov model (DBM) for simultaneous genotype calling and haplotype phasing in low-coverage NGS data of unrelated individuals. Our method is fully probabilistic that produces consistent inference of genotypes, haplotypes and recombination probabilities. Using data from the 1000 Genomes Project, we demonstrate that DBM not only yields more accurate results than some popular methods, but also provides novel characterization of haplotype structures at the individual level for visualization, interpretation and comparison in downstream analysis. DBM is a powerful and flexible tool that can be applied to many sequencing studies. Its statistical framework can also be extended to accommodate broader scopes of data.

Availability and implementation: http://stat.psu.edu/~yuzhang/software/dbm.tar

Contact: yuzhang@stat.psu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on September 25, 2012; revised on February 1, 2013; accepted on February 5, 2013

1 INTRODUCTION

Haplotype phasing is a long-standing problem in population genetics. The task is to computationally infer the combination of alleles (haplotype) at multiple single nucleotide polymorphisms (SNPs) on a single copy of chromosome, while the data are collected in genotype format (combination of alleles per SNP on two copies of chromosomes) from diploid genomes. Haplotype phasing is important because haplotypes provide rich information about the evolution history of individuals. Haplotype phasing is challenging because its complexity grows exponentially with respect to the number of SNPs. Although SNP arrays have been routinely used to collect genotype data from individuals, they only quantify genetic variants at known SNPs. Next-generation sequencing (NGS), on the other hand, has the ability to detect all genetic variants in individuals' genomes. Sequencing machine reads out genomes as short DNA fragments called 'reads'. After aligning the reads to a reference genome or by *de novo* assembly, putative SNPs are called if reads aligned at the same position carry alternative alleles. Genotypes are further called at the putative SNPs by comparing the read counts of alternative alleles. For haplotype phasing, SNPs are sorted by their positions so that information of linkage disequilibrium (LD) can be used. In this study, we will focus on NGS data, because genotype data is just a special case.

Many haplotype-phasing algorithms have been developed over the past decade (Browning and Browning, 2007; Howie *et al.*, 2009; Li and Stephens, 2003; Li *et al.*, 2009; Scheet and Stephens, 2006; Williams *et al.*, 2012). Recent methods all use pairs of Markov chains per diploid individual to model LD among SNPs. The states in Markov chains correspond to haplotypes, and the transitions of states indicate recombination events. Most methods, however, are designed for genotype data only. Although NGS data can be converted to genotypes before haplotype phasing, such approach will produce poor results in low coverage sequencing studies. Some methods (Howie *et al.*, 2009; Williams *et al.*, 2012) also require genetic maps, which are unavailable in many studies. We identified three methods that can phase haplotypes in NGS data: THUNDER (Li *et al.*, 2011), PPHS (Efros and Halperin, 2012) and HapSeq (Zhi *et al.*, 2012). THUNDER is a wrapper of MaCH (Li *et al.*, 2009), where the latter works on genotype data and is among the most accurate methods in haplotype phasing (Browning and Browning, 2011). We therefore used THUNDER as a benchmark in this study. PPHS uses perfect phylogeny to infer haplotypes. It, however, is designed for data in short regions without recombination, and relies on other methods to assemble short haplotypes into longer ones. We therefore did not include PPHS in this study. HapSeq is modified from THUNDER that improves on haplotype phasing if a read carries multiple alleles. We did not compare HapSeq, as it addresses a different problem. We included BEAGLE (Browning and Browning, 2007) as a second benchmark method. BEAGLE runs faster than THUNDER but performs as accurate (Browning and Browning, 2011). Although BEAGLE does not directly work on NGS data, it takes genotype likelihoods as input, which can be generated from read counts.

We introduce a Dynamic Bayesian Markov model (DBM) for phasing and characterizing haplotypes in NGS data of unrelated individuals. Three main reasons motivated this work. First, we want to design a coherent probabilistic model for haplotype phasing, such that inference consistency is theoretically justifiable. Some existing methods (Howie *et al.*, 2009; Li and Stephens, 2003; Li *et al.*, 2009; Scheet and Stephens, 2006)

use iterative conditional probabilities to infer haplotypes, for one individual at a time with the remaining individuals serving as donors. Such models do not have joint distributions corresponding to the conditionals used in iteration, and thus their results may not be consistent. Second, we want to develop a concise, but sufficient, representation of haplotypes for intuitive interpretation, visualization and comparison in downstream analysis. Existing methods only output strings of alleles as haplotypes, which neither reflect SNP dependencies nor suggest haplotype relationships among individuals. In many studies, haplotypes are not of direct interest. They are instead used as input in downstream studies such as association mapping and population inference. HaploView (Barrett *et al*., 2005) is one example of haplotype characterization, which shows haplotype block structures and SNP correlation. HaploView, however, does not characterize haplotypes at the individual level. Our third motivation is to provide users with a flexible tool for NGS data analysis, which can be applied to any species of interest with minimum input from the users.

A schematic view of our approach is shown in Figure 1. The observed read counts (Fig. 1a) do not carry haplotype information across SNPs. DBM takes read counts as input and infers haplotypes via three key components (Fig. 1b): an infinite-state hidden Markov model (HMM) modelling templates of haplotypes, with two Markov chains fitted to each individual; an emission probability of alleles from the state at each SNP; and an observation probability connecting alleles to the observed read counts. DBM is an example of non-parametric Bayes model (Dunson and Xing, 2009) equipped with Markov structures, and is also a variant of infinite-state HMM (Beal *et al*., 2002). We use Markov Chain Monte Carlo (MCMC) algorithms to estimate posterior distributions of model parameters of interest. Particularly, DBM outputs genotypes, haplotypes and recombination probabilities between SNPs. In addition, DBM produces segmentations of haplotypes at the individual level as mosaic combinations of states (Fig. 1c). Haplotypes within the same states are similar across SNPs, but not necessarily identical due

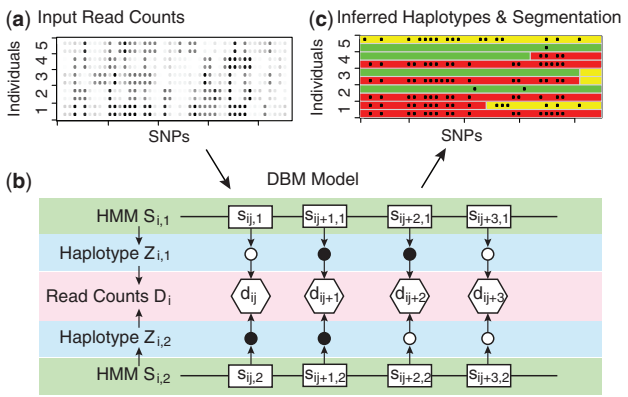

Fig. 1. Schematic view of DBM. (**a**) Input read counts of five individuals at 40 SNPs (grey scale of dots is proportional to the estimated mean number of minor alleles per SNP). (**b**) Each individual is fitted by two HMMs; alleles are generated independently by states; and read counts are generated by alleles (notations are defined in Section 2). (**c**) DBM inferred alleles (dots) and segmentation of haplotypes

to random mutations. The state information produced by DBM can be directly visualized to evaluate the relationships of haplotypes among individuals at the SNP resolution. Haplotype blocks and recombination hotspots can also be easily identified.

A challenging problem in HMM is to determine the number of states. Although too many states may reduce inference efficiency and over fit the data, too few states may not be sufficient to capture all the information in the data and thus loose power. In DBM, we allow the number of states to vary across SNPs. DBM uses a non-parametric Bayesian process to dynamically infer the number of states across SNPs. It has great flexibility to fit regions with either simple or complex structures. Simultaneously, DBM avoids over fitting the data via Bayesian regularization.

2 METHODS

2.1 Input data

DBM requires input of read counts of two alleles per putative SNP per individual. The SNPs should be ordered by their positions. DBM can also work for partially ordered (e.g. when reads are aligned to contigs) or unordered SNPs, but the accuracy of genotype calling and haplotype phasing will be affected due to loss of LD information (Nielsen *et al.*, 2011). Although DBM only considers biallelic SNPs, multi-allelic SNPs can always be converted to pseudo biallelic SNPs.

2.2 A dynamic Bayesian Markov model

Suppose that the NGS data (denoted by D) are collected from N individuals with L putative SNPs. For notation consistency, we use capital letters to denote an entire quantity and lower case letters to denote individual values. For example, d_{ij} denotes a pair of read counts observed from individual i at SNP j, and $D=\{d_{ij}\}$ denotes all data for $i=1,\ldots,N$ and $j=1,\ldots,L$.

For diploid genomes, we fit two HMMs per individual to model haplotype structures, where each HMM corresponds to one haplotype. Let $S=\{S_{i,k}\}$ denote the collection of HMMs, for $i=1,\ldots,N$ and $k=1,2$, with $S_{i,k}=(s_{i1,k},\ldots,s_{iL,k})$ denoting the states in one HMM across L SNPs, and $s_{ij,k}=1,2,\ldots$ taking positive integer values. In our model, states represent haplotype templates, and we allow infinite number of templates to be fitted to the data. For example, let $S_{i,k}=(1,1,1,3,3,2,2,2)$ denote the states of eight consecutive SNPs. It means that the corresponding haplotype is a concatenation of alleles from three templates: SNPs 1–3 carry alleles from template 1, SNPs 4–5 carry alleles from template 3 and SNPs 6–8 carry alleles from template 2. We assume that each state has its own allele distribution per SNP, and the alleles are independently generated from the states at each SNP.

To infer the state-specific allele distributions, we introduce an auxiliary variable $Z=\{Z_{i,k}\}=\{(z_{i1,k},\ldots,z_{iL,k})\}$, for $i=1,\ldots,N$ and $k=1,2$, with $z_{ij,k}=0$ or 1 indicating the presence of a 'minor' allele in the k-th haplotype of individual i at SNP j. Z represents the actual haplotypes in the sample. For instance, let $Z_{i,k}=(0,0,1,1,0,1,0,1)$ at eight SNPs. Combined with $S_{i,k}=(1,1,1,3,3,2,2,2)$, we may infer that template 1 emits alleles 0,0,1 at SNPs 1–3, respectively, template 3 emits alleles 1,0 at SNPs 4–5, respectively, and template 2 emits alleles 1,0,1 at SNPs 6–8, respectively. Given S and Z, we will not only learn the genotypes and the haplotypes in the sample, but also we can infer the state distributions and the state-specific allele distributions. We infer S and Z from the data using the following probabilistic model

$$\Pr(D,Z,S)=\Pr(D|Z)\Pr(Z|S)\Pr(S)= \left[\prod_{i=1}^{N}\prod_{j=1}^{L}\Pr(d_{ij}|z_{ij,1},z_{ij,2})\right]\times\left[\prod_{i=1}^{N}\prod_{j=1}^{L}\prod_{k=1}^{2}\Pr(z_{ij,k}|s_{ij,k})\right]\cdot \times\left[\prod_{i=1}^{N}\prod_{k=1}^{2}\Pr(s_{i1,k})\prod_{j=2}^{L}\Pr(s_{ij,k}|s_{i(j-1),k})\right] \quad (1)$$

Formula (1) has three components (Fig. 1b): the HMM $\Pr(S)$ of haplotype structures; the conditionally independent emission probability $\Pr(Z|S)$ of alleles given states; and the observation probability of read counts given alleles $\Pr(D|Z)$.

The observation probability $\Pr(D|Z)$ is a product of $\Pr(d_{ij}|z_{ij,1},z_{ij,2})$, where $(z_{ij,1},z_{ij,2})$ denote the two alleles at SNP j in individual i. Let $d_{ij}=(A_{ij},a_{ij})$ denote the read counts of the two alleles at SNP j in individual i. Let ε (default 0.01) denote the sequencing error rate. Let q_j (default 0.999) denote the probability that a putative SNP j is a true SNP, which can be calculated from its SNP quality score. Let $e_j=\min(\Sigma_i A_{ij},\Sigma_i a_{ij})/\Sigma_i(A_{ij}+a_{ij})$ denote the conditional probability of observing a wrong allele given that SNP j is a false-positive SNP. We write

$$\Pr(d_{ij}|z_{ij,1},z_{ij,2})=\Pr(A_{ij},a_{ij}|z_{ij,1},z_{ij,2}) \propto\begin{cases} q_j(1-\varepsilon)^{A_{ij}}\varepsilon^{a_{ij}}+(1-q_j)(1-e_j)^{A_{ij}}e_j^{a_{ij}}, & \text{if } z_{ij,1}+z_{ij,2}=0\\ 0.5^{A_{ij}+a_{ij}}, & \text{if } z_{ij,1}+z_{ij,2}=1\\ q_j\varepsilon^{A_{ij}}(1-\varepsilon)^{a_{ij}}+(1-q_j)e_j^{A_{ij}}(1-e_j)^{a_{ij}}, & \text{if } z_{ij,1}+z_{ij,2}=2 \end{cases}. \quad (2)$$

For homozygotes ($z_{ij,1}+z_{ij,2}=0$ or 2), formula (2) is a mixture model with two components corresponding to whether or not SNP j is a true SNP. For heterozygotes ($z_{ij,1}+z_{ij,2}=1$), SNP j is a true SNP and thus only has one component. The normalizing constants are not used in the model-fitting process and are ignored. More details of formula (2) can be found in Supplementary Material.

The emission probability $\Pr(Z|S)$ is modelled as a product of independent Bernoulli distributions conditioning on states. Each term $\Pr(z_{ij,k}|s_{ij,k})$ in formula (1) is a Bernoulli probability with state-specific 'minor' allele frequency $\{p^{(s,j)}\}$ in state s at each SNP j. Without knowing $\{p^{(s,j)}\}$, we assign a Dirichlet prior $Dir(\delta,\delta)$ and integrate $\{p^{(s,j)}\}$ out, where δ denotes a small constant (default $\max\{0.1,10/(\lambda N)\}$, and λ denotes the sequencing coverage). Let $n_0^{(s,j)}$, $n_1^{(s,j)}$ denote the number of haplotypes in state s carrying the major and the minor alleles at SNP j, respectively, we write

$$\Pr(Z|S)=\prod_{i=1}^{N}\prod_{j=1}^{L}\prod_{k=1}^{2}\Pr(z_{ij,k}|s_{ij,k}) =\prod_{j=1}^{L}\prod_{s=1}^{\infty}\frac{\Gamma(n_0^{(s,j)}+\delta)\Gamma(n_1^{(s,j)}+\delta)}{\Gamma(n_0^{(s,j)}+n_1^{(s,j)}+2\delta)}\frac{\Gamma(2\delta)}{\Gamma(\delta)\Gamma(\delta)} \quad (3)$$

In formula (3), the multiplication over states s can be computed in finite time, because $n_\cdot^{(s,j)}=0$ for all unoccupied states and hence their terms in (3) equal to 1. Derivation of formula (3) can be found in Supplementary Material.

Finally, the HMM $\Pr(S)$ is a product of $2N$-independent identically distributed Markov chains, with all chains governed by two sets of parameters: an infinite vector of state probabilities and a L-dim vector of SNP-specific recombination probabilities. Let $\{v_s\}$ denote the infinite vector of state probabilities that sum to 1. We use a stick-breaking process (Sethuraman, 1994) to describe the prior distribution of $\{v_s\}$. Let $\{V_s\}$ denote an infinite set of independent *Beta* random variables, $V_s\sim Beta(1,\alpha)$, with α denoting a hyper-parameter. We determine v_s by $v_s=V_s\prod_{t<s}(1-V_t)$. Using this prior, DBM allows and regularizes an infinite number of states in the model. The posterior distribution of $\{v_s\}$ is again a stick-breaking process. By default, we let $\alpha=1$. While larger α prefer more states to be fitted to the data, smaller α prefer fewer states.

To model the transition between states in our infinite-state HMM, we use a recombination mechanism: at each SNP j, the HMM decides whether or not to select a new state; if yes, a new state is randomly selected from distribution $\{v_s\}$, otherwise the state at SNP j remains the same as the state at SNP $j-1$. Let $\{r_j\}$ denote a L-dim vector of recombination probabilities at SNPs $j=1,\ldots,L$, with $r_1=1$ fixed. Let $\Phi=\{\Phi_{i,k}\}=\{(\phi_{i1,k},\ldots,\phi_{iL,k})\}$ denote the corresponding indicators of recombination events. We model $\phi_{ij,k}\sim\text{Bernoulli}(r_j)$ independently. The model for $(S_{i,k},\Phi_{i,k})$ is therefore written as

$$\Pr(S_{i,k},\Phi_{i,k})=\Pr(s_{i1,k})\prod_{j=2}^{L}\Pr(s_{ij,k},\phi_{ij,k}|s_{i(j-1),k}) =v_{s_{i1,k}}\prod_{j=2}^{L}(1-r_j)^{1-\phi_{ij,k}}(r_jv_{s_{ij,k}})^{\phi_{ij,k}}I^{1-\phi_{ij,k}}_{s_{i(j-1),k}=s_{ij,k}}$$

where the indicator $I^{1-\phi_{ij,k}}_{s_{i(j-1),k}=s_{ij,k}}$ equals to 0 if $\phi_{ij,k}=0$ and $s_{ij-1,k}\neq s_{ij,k}$. We assign a Dirichlet prior $Dir(\gamma,1-\gamma)$ and integrate $\{r_j\}$ out, with $0<\gamma\ll1$ denoting a small constant (default 0.01). Let $\xi_j=\sum_i\sum_k\phi_{ij,k}$ denote the total number of recombination events at SNP j in all Markov chains, we obtain (see Supplementary Material)

$$\Pr(S,\Phi)=\left[\prod_{i=1}^{N}\prod_{k=1}^{2}v_{s_{i1,k}}\prod_{j=2}^{L}I^{1-\phi_{ij,k}}_{s_{i(j-1),k}=s_{ij,k}}v^{\phi_{ij,k}}_{s_{ij,k}}\right] \times\left[\prod_{j=2}^{L}\frac{\Gamma(\xi_j+\gamma)\Gamma(2N-\xi_j+1-\gamma)\Gamma(1)}{\Gamma(2N+1)\Gamma(\gamma)\Gamma(1-\gamma)}\right] \quad (4)$$

Putting formulas (2–4) back to formula (1), along with the prior distribution of $\{v_s\}$ and the auxiliary variable Φ, we obtain the full DBM model in the form of $\Pr(D|Z)\Pr(Z|S)\Pr(S,\Phi|\{v_s\})\Pr(\{v_s\})$.

In summary, DBM is an infinite-state HMM with the states representing haplotype templates that emit alleles independently at each SNP. In turn, alleles generate the observed read counts. There are four sets of variables to be inferred from our model: (Z, S, Φ, $\{v_s\}$). Our HMM is specified by the initial distribution $\boldsymbol{v}=\{v_s\}$ and the transition matrix $\text{diag}(1-r_j,\infty)+r_j\mathbf{1}\boldsymbol{v}'$. Instead of estimating $\{r_j\}$, we introduce an auxiliary variable Φ and integrate $\{r_j\}$ out. The state variable S is a realization of HMMs regularized by Φ and $\{v_s\}$ with priors. Given S, alleles Z are generated independently at each SNP, which can also capture spurious mutations unexplained by LD. Although the dimensionality of these four sets of parameters is greater than the sample size, DBM is identifiably in a Bayesian framework. Particularly, the posterior distributions of the variables are balanced between the observations and the model priors. We next discuss our model inference using MCMC algorithms.

2.3 MCMC update

We infer DBM parameters iteratively using MCMC algorithms. Starting from a random initialization of model parameters, we use a forward-summation and backward-sampling algorithm to update (Z_i, Φ_i, S_i) for each individual i, conditioning on the parameters for the other individuals and $\{v_s\}$ in the current iteration. In the forward-summation step, we calculate the marginalized probability of data at SNPs $1,\ldots,j$, with parameters for SNPs $1,\ldots,j-1$ marginalized out via recursive summation, for $j=1,\ldots,L$ in ascending order, respectively. Marginalization is done over all possible states, recombination events and alleles at SNPs $1,\ldots,j-1$. To handle infinite number of possible states, we collapse states that are unoccupied in the current iteration into a 'super state', such that the total number of states in our calculation becomes finite. In the backward-sampling step, we then use the marginal probabilities to update $\{Z_{ij,k}\}$, $\{\Phi_{ij,k}\}$, $\{S_{ij,k}\}$, for $k=1,2$ and $j=L,\ldots,1$ in descending order, respectively. Sampling at SNP j is also conditioning on the states updated at SNP $j+1$. If a 'super state' is sampled, indicating an unoccupied state, we further sample an unoccupied state. To avoid local mode problems, we implemented additional MCMC updating schemes, including switching state labels and splitting states during burn-in. The detailed sampling procedures can be found in Supplementary Material.

Given the current states (S) and recombination events (Φ), we next update the state distribution $\{v_s\}$. Let $\{c_s\}$ denote the total number of state s selected at all recombination sites, i.e. at sites with $\phi_{ij,k}=1$. We first sample V_s from $V_s\sim Beta(c_s+1,\sum_{t>s}c_t+1+\alpha)$, the posterior distribution of V_s. We then calculate $v_s=V_s\prod_{t<s}(1-V_t)$. We only calculate v_s for a

finite number of states up to state s^*, where s^* denotes the maximum state index in S in the current iteration, because we collapse the unoccupied states with indices $>s^*$ into a super state during MCMC, the probability of which is $1-\sum_{s\le s^*} v_s$.

We repeat the above updating procedures many times and then collect posterior samples of (Z, S, Φ). The output of DBM includes the inferred haplotypes (and genotypes and SNP calls), the recombination probabilities at each SNP and the underlying haplotype structures. To determine the final haplotypes, we first use maximum *a posteriori* (MAP) to call genotypes from the posterior samples of Z, at each SNP for each individual separately. We then slide a five-heterozygote window across all detected heterozygotes in each individual to determine their haplotype configurations. Starting from the first five heterozygotes, we use MAP to determine their joint haplotype configurations. We then slide the window to the right by one heterozygote, and we use MAP to determine the configuration of the new heterozygote conditioning on the haplotype configurations of the other four heterozygotes in the window. We repeat this procedure across all heterozygotes to obtain the entire haplotype pair for each individual. Direct MAP of the entire haplotypes is computationally intractable. Using this procedure, we can recover haplotype information accurately and efficiently. When summarizing haplotypes, we further determine their underlying states from the posterior samples of states (S). Conditioning on the haplotype pair determined in the current window, we identify all posterior samples of states carrying the haplotype pairs. We then use MAP to determine the state configuration for both heterozygotes and homozygotes. Finally, we estimate the recombination probabilities from the posterior samples of Φ by calculating the proportion of recombination events occurred at each SNP.

3 RESULTS

3.1 Simulation from human data

We evaluated the performance of DBM using datasets generated from human sequences with European (CEU) and African (YRI) origins. We downloaded the phased haplotypes of CEU and YRI individuals from the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2010). Using these reference haplotypes, we simulated haplotypes of new individuals and their corresponding read counts as follows: (i) we generated new haplotypes as mosaic combination of reference haplotypes with transition rate 1 per 200 kb, and reference individuals are randomly chosen; (ii) we randomly paired new haplotypes to form new individuals; (iii) we simulated read counts at each SNP using a Poisson distribution with mean $\lambda x/2$, where λ denotes the sequencing coverage and $x=0,1,2$ denotes the allele count; (iv) we generated random sequencing errors in read counts across the genome at rate 0.01 per basepair per read; this created both false-positive alleles and false-positive SNPs; and (v) we removed all reads carrying alleles that are different from the two most frequent alleles at each SNP, as they are most likely sequencing errors; we also removed SNPs whose minor read count is less than a threshold, such that the total number of false-positive SNPs in the data is controlled <5%.

3.2 Accuracy in genotyping and haplotype phasing

We ran DBM on the simulated datasets with sequencing coverages $\lambda = 1.0$, 3.0 and 6.0, and sample sizes $N=10$, 20, 40, 80, 160, 320, 640, respectively. Each dataset contained 10 000 SNPs from a randomly chosen genomic region. We compared DBM with THUNDER (Li *et al*., 2011) and BEAGLE (Browning and Browning, 2007). We ran THUNDER by its default setting for 100 iterations with sequencing error rate specified at 0.01. For computing speed, the maximum number of states used by THUNDER is bounded by 200. We also run DBM and BEAGLE for 100 iterations. BEAGLE does not take read counts. We therefore input BEAGLE with the genotype likelihoods generated by DBM. To evaluate the benefit of using LD in genotype calling, we further implemented a single SNP call method, which determines genotypes at each SNP separately. This is done by fixing the recombination probability at 1 at all SNPs in DBM. In practice, one may call genotypes first and then make a bona fide use of the called genotypes to infer haplotypes. To evaluate the power of this two-step approach, we ran MaCH on the called genotypes generated by the single SNP call method.

We used two accuracy measures to compare the results, one for genotype calling (percentage of incorrect alleles) and one for haplotype phasing (percentage of switch errors). For genotype calling, we calculated the number of alternative alleles (relative to an arbitrarily chosen reference allele), denoted by x_{ij}, estimated by each program at SNP j in individual i. Further, we denote the true number of alternative alleles by g_{ij}. The percentage of incorrect alleles is defined as $\sum_{ij}|x_{ij}-g_{ij}|\ /\ (2NL)$. For haplotype phasing, we first calculated the number of switches needed to convert the inferred haplotypes to the true haplotypes in each individual. We then divided that number by the number of heterozygous SNPs in the individual minus 1. We only used the correctly inferred heterozygous SNPs by each program, so that the results were not strongly affected by genotyping errors. Finally, the overall switch error is averaged across all individuals.

Figure 2 shows the genotyping accuracy of the four programs. At low sequencing depth ($\lambda=1$), DBM performed consistently the best among all methods, especially in small samples. In contrast, THUNDER performed slightly worse than DBM, but BEAGLE performed poorly in small samples. It was as inaccurate as the single SNP call method at $N=20$. At larger sequencing depths ($\lambda=3$ or 6), the performance of the first three programs became much more similar in both CEU and YRI individuals, except that BEAGLE performed slightly worse than DBM and THUNDER at $N=10$. All three programs substantially outperformed the single SNP call method, suggesting that LD can greatly help improving the accuracy of genotype calling (Nielsen *et al.,* 2011).

We next show in Figure 3 the haplotype phasing result. At all sequencing depths ($\lambda=1,3,6$), DBM performed consistently the best in small samples (e.g. $N\le 80$). BEAGLE performed worse than both DBM and THUNDER at $\lambda=1$, which may be related to its erroneous genotype calls. THUNDER performed consistently and substantially worse than DBM in small samples (e.g. $N\le 80$) regardless of the sequencing depth. In large samples ($N\ge 320$), DBM performed slightly worse than BEAGLE and THUNDER, which may be due to the local mode problem in the MCMC algorithm. Given the dynamic nature of DBM that selects varying numbers of states to fit the data, the method may be trapped in a suboptimal mode when many individuals are fitted simultaneously. In such cases, multiple independent runs of DBM from different starting values and advanced MCMC sampling techniques (Liu, 2001) may be desirable to improve its performance. Finally, the two-step approach of single SNP call + MaCH phasing performed the worst in most cases.

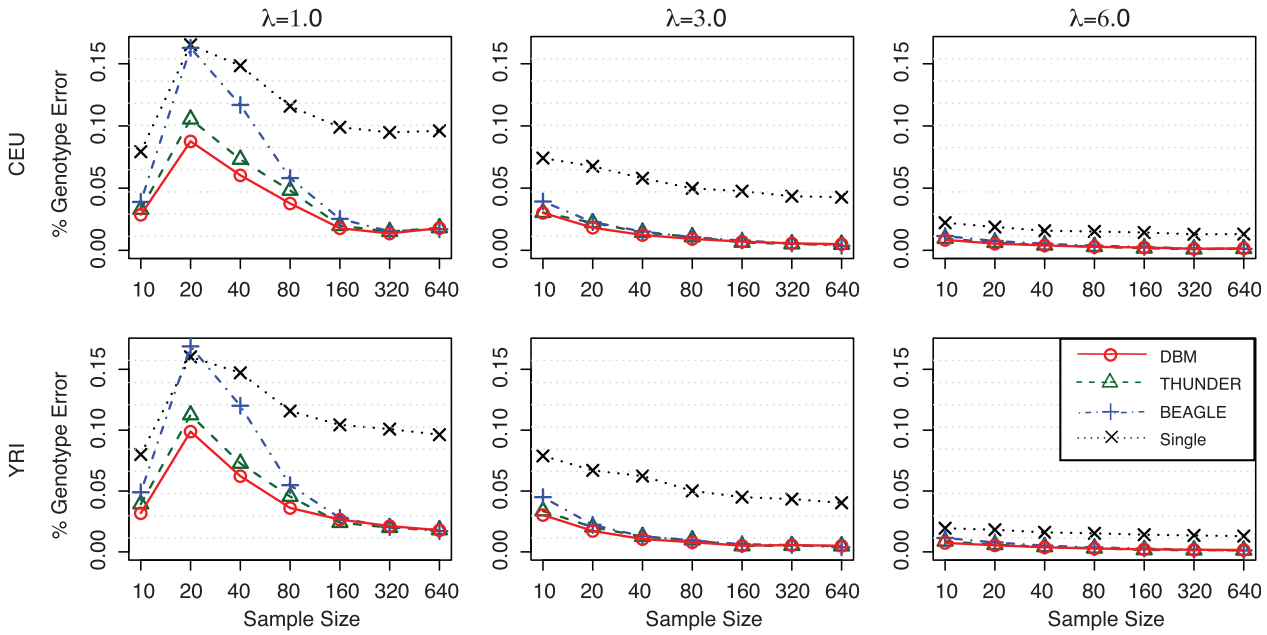

Fig. 2. Genotyping accuracy in CEU and YRI samples at different sequencing depth (λ) by DBM, THUNDER, BEAGLE and Single SNP Call. Sample size is shown in log scale

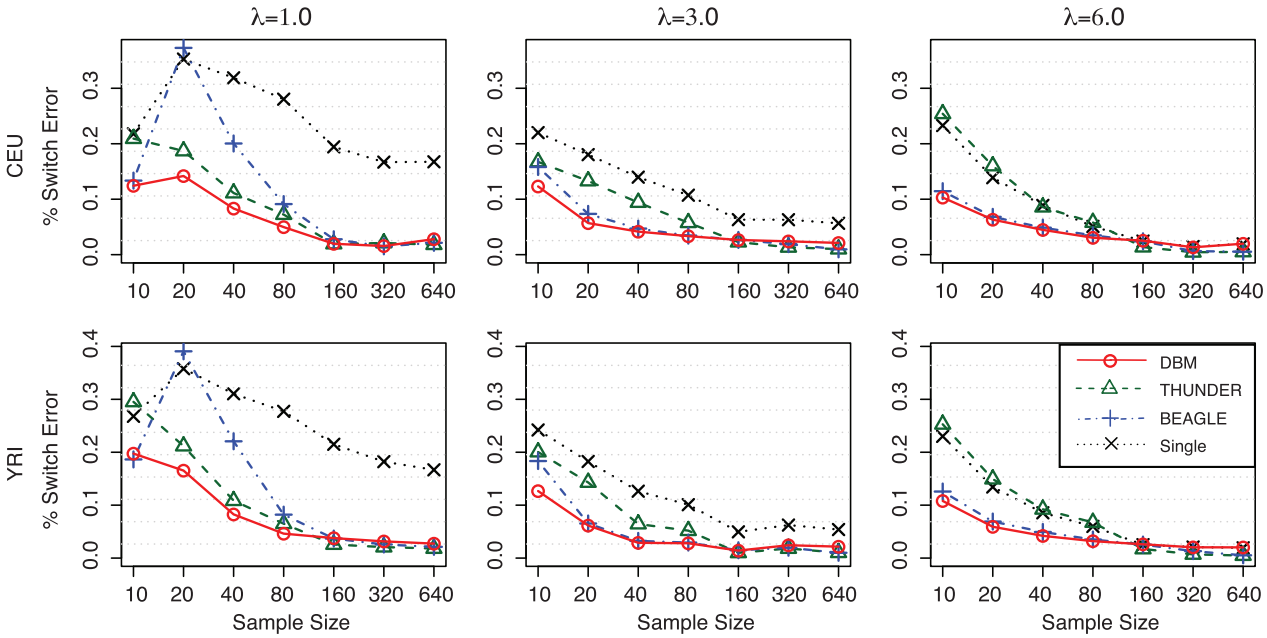

Fig. 3. Haplotype phasing accuracy in CEU and YRI samples at different sequencing depth (λ) by DBM, THUNDER, BEAGLE and a two-step approach of Single SNP Call followed by MaCH phasing. The sample size is shown in log scale

3.3 Run time

Figure 4 shows the computing time of the three programs for the simulated CEU datasets at $\lambda = 3$. At $N = 10$, DBM was the slowest program among the three. At $N \geq 20$, however, DBM ran up to 8 × faster than THUNDER. Both DBM and THUNDER ran in time complexity $O(|S|^2 NL)$, i.e. proportional to the square of the number of states and linear to the number of individuals and SNPs. Since DBM dynamically selects the number of states to fit the data, the number of states used by DBM can be far less than that used by THUNDER in large samples, which then can greatly improve the computation speed. Interestingly, while BEAGEL ran the fastest among the three programs in small samples ($N \leq 160$), its computing speed does not scale up well in large samples. Particularly, BEAGEL ran slower than DBM at $N \geq 320$ in this study. In Figure 4, we also observed that the computing time of DBM is almost linear with respect to the

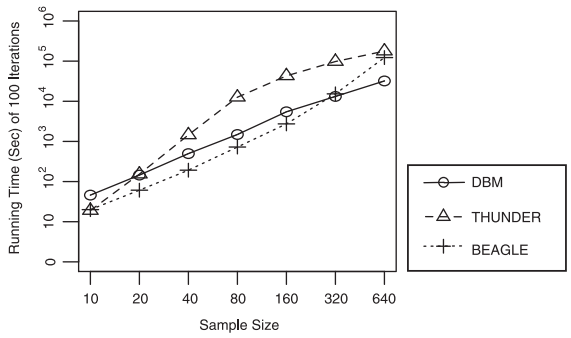

Fig. 4. Computing time of DBM, THUNDER and BEAGLE on CEU datasets with sequencing depth $l = 3$ at 10 000 SNPs. Time and sample size are both shown in log scale

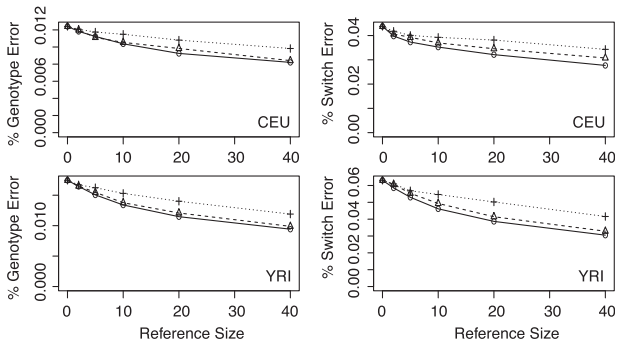

Fig. 5. Performance of DBM using different types of reference in different sizes. Left: genotyping error rate. Right: haplotype phasing switch error rate. Reference data are input in three ways: haplotypes (solid line), genotypes (dashed line), read counts (dotted line)

sample size. The computing time of THUNDER is theoretically cubic to the sample size, but due to the fact that its maximum number of states is bounded at 200, its computing time is linear at $N > 100$. The computing time of BEAGLE appears to be non-linear with respect to the sample size. As previously reported (Williams *et al.*, 2012), BEAGLE's running time grows faster than the sample size.

3.4 Using reference input

DBM can further take input of reference genotypes and haplotypes. To use reference genotypes, one can simply convert genotypes to pseudo read counts. For genotypes AA, Aa and aa, the read counts can be written as $(2X,0)$, (X,X) and $(0,2X)$, respectively, with large X (e.g. ≥ 15). If the reference data are in haplotype format, we treat each reference haplotype as an 'individual', and we fit each reference haplotype with one Markov chain only. To do this, we replace the observation probability $\Pr(D|Z)$ in formula (2) by an indicator function $I_{z=a}$ at each SNP, with 'a' denoting the true allele in the current haplotype at each SNP.

To evaluate the benefit of using extra data in genotype calling and haplotype phasing, we simulated 40 individuals (CEU and YRI, respectively) at 10 000 SNPs with sequencing depth $\lambda = 3$ and sequencing error rate 0.01. These are the sample individuals. We further simulated additional 40 individuals (CEU and YRI, respectively) as the references. We ran DBM to analyse the 40 sample individuals along with various numbers of references input to the program. The references were input in three ways: (i) in form of read counts at sequencing depth $\lambda = 3$, which then merely increased the sample size; (ii) in form of known genotypes, which eliminated genotyping uncertainties in the reference data; and (iii) in form of known haplotypes, which further provided phasing information in the reference data. As shown in Figure 5, DBM can indeed gain power in genotype calling and haplotype phasing from the reference data. By comparing the three types of reference input, reference haplotypes provided the largest accuracy boost in both CEU and YRI samples. In practice, however, haplotypes are expensive to obtain. Alternatively, reference genotypes also significantly improved the accuracy of genotyping and haplotype phasing, just slightly worse than using reference haplotypes, but clearly better than merely increasing the sample size.

3.5 Haplotype characterization

Our method further produces haplotype segmentation that captures the allele compositions and dependencies. Haplotype segmentation is specified by the recombination indicator Φ in DBM that partitions each haplotype into consecutive intervals, and also by the state variable S in DBM that specifies the haplotype template index of each interval. Since haplotypes in the same templates at the same SNPs have similar allele compositions, the haplotype segmentation output by DBM is useful in downstream analysis, e.g. for population structure inference and association studies. An example of haplotype segmentation is shown in Figure 6a, a mixture of 10 CEU and 10 YRI individuals from 1000 Genomes at 2000 SNPs. DBM segmented the 20 individuals using 14 states as shown in colours. Relationships among individuals can be clearly seen: the states shared by the CEU individuals are quite different from the states shared by the YRI individuals. Recombination hotspots can also be observed at SNPs with frequent transitions between states. In association studies, one can use the haplotype segmentation in similar ways to visually identify associated loci among individuals ascertained by phenotypes. Formal test on the association of haplotype segments is also possible.

DBM outputs haplotype templates as summary statistics from the data. The haplotype templates can be intuitively treated as a reconstruction of the ancestral haplotypes, although not exactly so because our model does not involve a time component. In Figure 6b, we show the inferred haplotype templates in CEU and YRI samples, respectively. Each colour represents one template (the colours match with those shown in Fig. 6a), and the height of colour bars indicates their estimated population abundance at each SNP. Darkness of colours is drawn proportionally to the probability that the SNP carries a minor allele in the corresponding template. Comparing the haplotype templates in CEU and YRI, we observed that the two sets of individuals only shared small proportions of genetic contents (templates) at each SNP, and hence they are genetically separable. The YRI samples carried more diverse genetic contents than the CEU samples. Also, the proportions of templates varied across regions, reflecting genetic variability. Figure 6c shows the number of distinct states fitted to the data by DBM, where the number of states fitted to the YRI sample is consistently larger

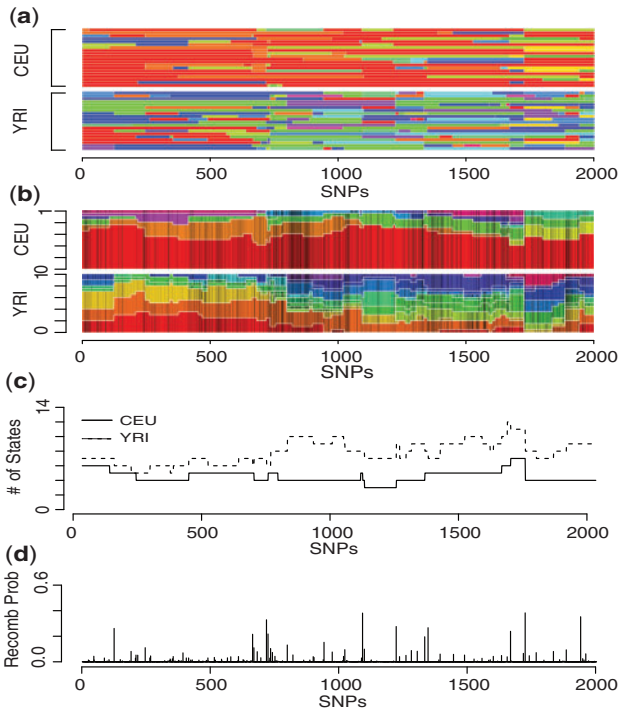

Fig. 6. Comparing CEU and YRI samples using DBM. (**a**) Haplotype segmentation of 10 CEU and 10 YRI individuals; each row corresponds to one chromosome, and each colour corresponds to a state. (**b**) Reconstructed haplotype templates with the same colours as used in (a); each colour represents one haplotype template over 2000 SNPs; at each SNP, the height of each colour bar represents the population proportion of the template; colour darkness at each SNP is proportional to the alternative allele frequency. (**c**) Number of distinct haplotype templates used at each SNP in CEU and YRI samples. (**d**) Inferred recombination probabilities

than the number of states fitted to the CEU sample, again indicating greater genetic diversity of YRI than CEU. Finally, Figure 6d shows the estimated recombination probabilities between SNPs. This example illustrates the utility of DBM characterization of haplotypes for evaluating genetic relatedness and diversity among individuals at the SNP resolution.

4 DISCUSSION

We introduced a dynamic Bayesian Markov model for joint inference of genotypes and haplotypes in NGS data. The method is fully probabilistic and produces consistent inference results. A main feature of DBM is its infinite-state Markov chain that allows varying numbers of mixture components fitted to the data depending on the structural complexity of the data across regions, such that haplotype structures and SNP dependencies can be most efficiently and sufficiently captured by the states. Using data from the 1000 Genomes Project with individuals of different ethnicity, we compared DBM with two popular algorithms: THUNDER and BEAGLE, as they both have been used to phase haplotypes in the 1000 Genomes Project (the 1000 Genomes Project Consortium, 2010). In all scenarios tested, DBM produced either similar or better results than the two benchmark programs. Particularly, for small sample datasets and/or low sequencing studies, DBM performed substantially better than the other two programs. DBM is thus desirable for exploratory sequencing studies that involve limited samples at low sequencing coverage. For large sample datasets, DBM performed similarly to the other two methods, and DBM had better runtime scalability with respect to the sample size. All three programs tested in this study call genotypes and phase haplotypes simultaneously, which is more powerful than the two-step approach. Consistent with previous reports (Nielsen *et al.,* 2011), we demonstrated that using LD information can substantially increase the accuracy of genotype calling.

DBM is a flexible tool that can be applied to many sequencing projects with minimum input requirement from the user. DBM takes input of either read counts or genotypes (in form of pseudo read counts), and outputs genotypes, haplotypes and recombination probabilities. Additional information such as SNP quality scores and genetic maps can also be provided, but not mandatory. Missing genotypes can be easily accommodated and imputed by specifying zero read counts for both alleles in the input file. Although all examples shown in this article were generated from data in human genome, DBM can be directly applied to sequencing studies of any diploid species. To fully use LD information, SNPs should be ordered by their genomic positions. On the other hand, DBM can work on unsorted SNPs, in which case it reduces to a single SNP call method.

DBM characterizes haplotypes via segmentation that captures the haplotype relationships among individuals and de-correlates alleles across SNPs. It also reveals the most likely recombination loci at the individual level. One can use DBM segmentation in downstream analysis for hypothesis testing and parameter inference, such as association mapping and population evolution studies. Most *de novo* population detection algorithms require independent SNPs, whereas our approach enables use of all SNPs in form of haplotype segments, which can potentially significantly increase the power for detecting subtle stratification.

DBM reports summary statistics of a sample in form of haplotypes templates. The templates are identified as commonly shared haplotypes among individuals. The haplotype templates can be used to learn genetic relatedness and diversities within and between groups of samples at the SNP resolution. To compare groups of individuals, data from all groups of individuals should be input to DBM together, such that the state indices are matched across groups.

Population NGS data are being increasingly generated in many species of interest, and the sequencing cost continues to drop. Although DBM only facilitates the first step in population sequencing studies, the DBM model itself has broader applications. For example, we can modify the observation and emission probability functions to take various types of data into account. We can also modify the DBM model for *de novo* detection of population stratification and admixture mapping, where we allow unknown numbers of populations to be admixed without requiring ancestral references.

ACKNOWLEDGEMENTS

The author is grateful to the two anonymous reviewers for their constructive comments that have substantially improved the quality of this manuscript.

Funding: The author acknowledges the grant support of NIH R01HG004718 and NIH 1UL1RR033184.

Conflict of Interest: none declared.

REFERENCES

Barrett,J.C. *et al.* (2005) Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, **21**, 263–265.

Beal,M.J. *et al.* (2002) The infinite hidden Markov model. In: *Advances in Neural Information Processing Systems 14*. The MIT Press, Cambridge, MA, USA, pp. 577–584.

Browning,S.R. and Browning,B.L. (2007) Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am. J. Hum. Genet.*, **81**, 1084–1097.

Browning,S.R. and Browning,B.L. (2011) Haplotype phasing: existing methods and new developments. *Nat. Rev. Genet.*, **12**, 703–714.

Dunson,D.B. and Xing,C.H. (2009) Nonparametric Bayes modeling of multivariate categorical data. *J. Am. Stat. Assoc.*, **104**, 1042–1051.

Efros,A. and Halperin,E. (2012) Haplotype reconstruction using perfect phylogeny and sequence data. *BMC Bioinformatics*, **13** (**Suppl. 6**), S3.

Howie,B.N. *et al.* (2009) A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *Plos Genet.*, **5**, e1000529.

Liu,J.S. (2001) *Monte Carlo Strategies in Scientific Computing*. New York, Springer.

Li,N. and Stephens,M. (2003) Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, **165**, 2213–2233.

Li,Y. *et al.* (2009) MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet. Epi.*, **34**, 816–834.

Li,Y. *et al.* (2011) Low-coverage sequencing: implications of design of complex trait association studies. *Genome Res.*, **21**, 940–951.

Nielsen,R. *et al.* (2011) Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.*, **12**, 443–451.

Scheet,P. and Stephens,M. (2006) A fast and flexible statistical model for large scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am. J. Hum. Genet.*, **78**, 629–644.

Sethuraman,J. (1994) A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, **4**, 639–650.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

Williams,A.L. *et al.* (2012) Phasing of many thousands of genotyped samples. *Am. J. Hum. Genet.*, **91**, 239–251.

Zhi,D.G. *et al.* (2012) Genotype calling from next-generation sequencing data using haplotype information of reads. *Bioinformatics*, **28**, 938–946.