Title
Fast and accurate site frequency spectrum estimation from low coverage sequence data

Permalink
https://escholarship.org/uc/item/104381sb

Journal
Bioinformatics, 31(5)

ISSN
1367-4803

Authors
Han, E
Sinsheimer, JS
Novembre, J

Publication Date
2015-03-01

DOI
10.1093/bioinformatics/btu725

Peer reviewed
Genetics and population analysis

Fast and accurate site frequency spectrum estimation from low coverage sequence data

Eunjung Han\(^1\)*, Janet S. Sinsheimer\(^1,2\) and John Novembre\(^3\)*

\(^1\)Department of Biostatistics, University of California, Los Angeles, Los Angeles, CA 90095, USA, \(^2\) Department of Human Genetics and Biomathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA and \(^3\)Department of Human Genetics, University of Chicago, Chicago, IL 60637, USA

*To whom correspondence should be addressed.

Associate Editor: Jeffrey Barrett

Received on June 24, 2014; revised on October 17, 2014; accepted on October 27, 2014

Abstract

**Motivation:** The distribution of allele frequencies across polymorphic sites, also known as the site frequency spectrum (SFS), is of primary interest in population genetics. It is a complete summary of sequence variation at unlinked sites and more generally, its shape reflects underlying population genetic processes. One practical challenge is that inferring the SFS from low coverage sequencing data in a straightforward manner by using genotype calls can lead to significant bias. To reduce bias, previous studies have used a statistical method that directly estimates the SFS from sequencing data by first computing site allele frequency (SAF) likelihood for each site (i.e. the likelihood a site has each possible allele frequency conditional on observed sequence reads) using a dynamic programming (DP) algorithm. Although this method produces an accurate SFS, computing the SAF likelihood is quadratic in the number of samples sequenced.

**Results:** To overcome this computational challenge, we propose an algorithm, ‘score-limited DP’ algorithm, which is linear in the number of genomes to compute the SAF likelihood. This algorithm works because in a lower triangular matrix that arises in the DP algorithm, all non-negligible values of the SAF likelihood are concentrated on a few cells around the best-guess allele counts. We show that our score-limited DP algorithm has comparable accuracy but is faster than the original DP algorithm. This speed improvement makes SFS estimation practical when using low coverage NGS data from a large number of individuals.

**Availability and implementation:** The program will be available via a link from the Novembre lab website (http://jnpopgen.org/).

**Contact:** ehan416@gmail.com, jnovembre@uchicago.edu

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

1 Introduction

A site frequency spectrum (SFS) describes the distribution of allele frequencies across sites in the genome of a particular species. The SFS is of primary interest in population genetics, as it is a complete summary of sequence variation at unlinked sites and its shape reflects underlying population genetic processes, such as growth, bottlenecks and selection. Moreover, a number of population genetic inferences can proceed directly from the SFS. For example, demographic history (e.g. evidence for population expansions, bottlenecks or migrations) can be directly inferred from the SFS [using, for example, dadi (Gutenkunst et al., 2009) or (Excoffier et al., 2013)]. The SFS can also be compressed down to univariate summary statistics that form the basis of popular neutrality tests (Achaz, 2008, 2009; Fay and Wu, 2000; Fu and Li, 1993; Tajima, 1989) that underlie many empirical genome-wide selection scans (e.g. Andolfatto, 2007; Begun et al., 2007). Hence, inferring the precise SFS from genetic data is crucial in many population genetic analyses.
With the recent rapid progress in sequencing techniques, obtaining large-scale genomic data from thousands to tens of thousands of individuals is practical (e.g., 1000 Genomes Project Consortium, 2010, 2012; Fu et al., 2013; Nelson et al., 2012) and this increased sample size enables us to conduct more accurate population genetic inference. However, current massively parallel short-read sequence technologies also pose many inherent challenges—for example, reads have high error rates, read mapping is sometimes uncertain and coverage is variable and in many cases low or completely absent. These challenges make accurate individual-level genotype calls difficult and make some downstream analysis based on the inferred genotypes problematic.

In a previous study (Han et al., 2014), we showed that the SFS computed from genotype calls (a call-based estimation approach) is biased at low to medium coverage (<10×), whereas the SFS directly inferred from aligned short-read sequencing data (a direct estimation approach) is unbiased even at low coverage. The direct estimation approach infers the maximum likelihood estimate (MLE) of the SFS by an EM algorithm (Li, 2011) or a Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Nielsen et al., 2012), assuming independence across all individuals and sites. Both of these algorithms are implemented in the ANGSD software package (Nielsen et al., 2012).

Both of these algorithms require computation of the site allele frequency (SAF) likelihood for all sites. These vectors contain the likelihood that an allele for each possible allele frequency at a site (regardless whether monomorphic or polymorphic) conditional on observed sequence reads. Based on the precomputed SAF likelihoods, the MLE of the SFS is obtained by optimization, using either the EM (Li, 2011) or the BFGS algorithm (Nielsen et al., 2012). The bottleneck in obtaining the MLE of the SFS is computing the SAF likelihoods, rather than optimization. In fact, the maximization of the likelihood either by the EM or the BFGS algorithm takes only a small fraction of time compared with the computation of the SAF likelihood. This is because computation of the SAF likelihood at each site requires a summation over all possible genotype combinations for n individuals and naive computation of this sum has a runtime complexity of O(3n). To overcome this computational burden, Li (2011) proposed a dynamic programming (DP) algorithm to effectively compute the SAF likelihood for each site in O(n2) and Nielsen et al. (2012) implemented this algorithm in the ANGSD software. However, this algorithm is still not practical to use if there are large numbers of individuals, because it is quadratic in the number of genomes (see Fig. 1B for runtime). Moreover, this algorithm is numerically unstable for a large sample (Li, 2011). To solve this problem of computational inefficiency and numerical instability, we compute the SAF likelihood in a more efficient way that still retains the accuracy of the original DP algorithm. Our new method uses a combination of rescaling and sensible approximation to compute the SAF likelihood.

2 Approach

To establish notation and background, we first review the existing DP algorithm implemented in the ANGSD software (Nielsen et al., 2012) and then introduce our approach.

2.1 DP algorithm used by ANGSD

Let D denote the short-read sequencing data and X represent a total count of the derived allele for a sample of n diploid individuals at a particular site. The corresponding SAF likelihood, $h = (h_0, h_1, \ldots, h_{2n})$, is a $(2n + 1)$-dimensional vector in which each element, $h_j = P(D|X = x)$, is the likelihood that the derived allele frequency in the sample is $x/(2n)$:

$$h_x = \frac{1}{(2n)!} \sum_{x=0}^{2n} \sum_{g=0}^{2n} \prod_{k=1}^{n} \binom{g_k}{2} L_k(g_k),$$  (1)

where $I()$ is an indicator function and $L_k(g_k) = P(D_k|G_k = g_k)$ is a genotype likelihood of the individual k for genotype $g_k$.

To calculate the SAF likelihood $h$ by the DP algorithm, define a raw SAF likelihood for $j$ individuals ($(2j + 1)$-dimensional vector), given by $x^j = (z^j_0, z^j_1, \ldots, z^j_{2j})$, in which each element is defined as

$$z^j_k = \sum_{g=0}^{2n} \sum_{h=0}^{2n} \prod_{t=1}^{j} \binom{g_t}{2} L_t(g_t),$$  (2)

where $j = 1, \ldots, n$ and $x = 1, \ldots, 2j$. Note that this expression does not include a rescaling factor $\frac{2n}{x}$.

The vector $x^j$ is iteratively updated from the vector $x^{j-1}$ (raw SAF likelihood for $j - 1$ individuals) by the following recurrence relation:

$$z^j_k = L_j(0) z^{j-1}_k + 2L_j(1) z^{j-1}_{k-1} + L_j(2) z^{j-1}_{k-2}. $$  (3)

In a final step, each element of the vector $x^n$ is rescaled by a corresponding factor $\frac{2n}{x}$ to obtain the vector $h$ (i.e., $h_x = z^n_x / \left(\frac{2n}{x}\right)$), and then the resulting vector $h$ is standardized such that the maximum element of the vector becomes 1, as likelihoods need only be defined proportional to a constant.

To illustrate the procedure, we show how the raw SAF likelihood is recursively updated from $x^1$ to $x^n$ by the DP algorithm in the ANGSD software. Each row in a lower triangular matrix in Figure 2A (top) represents the raw SAF likelihood for $j$ individuals (a vector of length $2j + 1$). Figure 2A also shows how the raw SAF likelihood $x^n$ (middle) is converted to the final SAF likelihood $h$ (bottom) after rescaling by $\left(\frac{2n}{x}\right)^{-1}$ and standardization.
In this example, it requires computation of 2600 elements (because shows the final SAF likelihood of which the range goes from 0 to 2000. Middle and bottom rows show how the SAF likelihood is recurrently updated by the original DP algorithm. Each row in a low triangular matrix represents the raw SAF likelihood for j individuals (x). The value of the SAF likelihood is shown in a color scale, of which the range goes from 0 to 2000. Middle and bottom rows show how the raw SAF likelihood x (the last row of the lower triangular matrix) is converted to the final SAF likelihood b by rescaling and standardization. Note that the final SAF likelihood b has a peak at the derived allele count of 0. (B) Top row shows how the rescaled SAF likelihood is recurrently updated by the rescaled DP algorithm. Each row in the lower triangular matrix represents the rescaled SAF likelihood b’ for j individuals. The value of the SAF likelihood is shown in a color scale, of which the range goes from 0 to 1. Bottom row shows the final SAF likelihood b. In both (A) and (B), the gray area represents the range of values computed by the original and the rescaled DP algorithm. In this example, it requires computation of 2600 elements (because 3 + 5 + · · · + 101 = 350 + 2i + 1) to update the SAF likelihood.

2.2 Rescaled DP algorithm

In our preliminary work, we observed that the value at the mode of x is relatively large. In this example with 50 diploid individuals, the mode of x is 1973. With 500 diploid individuals, the mode of x can be about 8 × 10^{12} (data not shown). This implies that the DP algorithm can have an arithmetic overflow problem (i.e. a computed value is greater in magnitude than the largest value that a computer can store in memory) for large samples because the mode of x increases exponentially as the sample size increases. Furthermore, the values at the edges of x are very small. In this example with 50 individuals, the value of the SAF likelihood function for the derived allele count of 100 is 7 × 10^{-10}. With 500 diploid individuals, the value of the SAF likelihood function for the derived allele count of 1000 is smaller than 10^{-300} (data not shown). This implies that the DP algorithm can have an arithmetic underflow problem (i.e. a computed value is smaller in magnitude than the smallest value that a computer can store in memory) for large samples because the values at the edges of x keep decreasing exponentially as sample size increases. Therefore, the original DP algorithm will be numerically unstable for large samples, as the computation of produces both numerical overflow (at the mode of x) and underflow (away from the mode).

To overcome the numeric instability of the DP algorithm, we modified the original DP algorithm such that rescaling and standardization take place at each step of updating the SAF likelihood. For this modified algorithm, we define a rescaled SAF likelihood for j individuals ((2j + 1)-dimensional vector), b’ = (b’_0, b’_1, · · · , b’_{2j}), of which each element is defined as

\[ b'_k = \frac{1}{2j+1} \sum_{x=0}^{2j} \sum_{g=0}^{2j} P(x) \left( \prod_{k=1}^{2j} P(g)_k \right), \]  

where \( j = 1, \ldots, n \) and \( x = 1, \ldots, 2j \). We can derive a recurrence relation to iteratively update the vector b’ from the vector b’⁻¹ (rescaled SAF likelihood for j – 1 individuals) as follows:

\[ b'_k = \frac{2(j - 1)}{x} L_x(0) b'_{x-1} + 2 \frac{(x - 2j - 1) L_x(1) b'_{x+1}}{x} + 2 \frac{(j - 1) L_x(2) b'_{x+2}}{x} - \frac{2}{x^2} \]

Because the constant \( \frac{1}{2j+1} \) in Equation (5) is cancelled out during standardization, we can use the following recurrence equation to update the rescaled SAF likelihood:

\[ b'_x = \left( (2j-x)(2j-x-1)L_x(0)b'_{x-1} + 2x(2j-x)L_x(1)b'_{x+1} + x(x-1)L_x(2)b'_{x+2} \right) \]

for \( x = 1, \ldots, 2j \).

Figure 2B shows how the SAF likelihood is recurrently updated from h to h by the rescaled DP algorithm in a lower triangular matrix (Figure 2B, top) and the SAF likelihood h (Figure 2B, bottom). Now, all values in the SAF likelihood h range between 0 and 1, suggesting there will be no potential numerical overflow. Importantly, we observed that the most of the cells in the SAF likelihood during update have a value close to 0 (shown in gray). This implies that computing all values of the SAF likelihood is inefficient and we can accurately approximate this vector by only computing the first few elements and setting the rest of the elements to 0. This motivated the development of the banded and score-limited DP algorithm.

2.3 Banded and score-limited DP algorithm

We observed that all non-negligible values of the updated rescaled SAF likelihoods, h to h, are consistently concentrated on the best-guess allele counts (Figure 2 for the allele frequency of 0 and Figure 3 for the allele frequency of 0.7). For example, for a site that is fixed for the ancestral allele, we observe that all non-negligible values of the rescaled SAF likelihoods are consistently observed on the first few cells of the vector, and the final SAF likelihood h has a peak at the allele frequency of 0 (Fig. 2B). For a site that is polymorphic, we observe that the mode of the SAF likelihood typically stays at 0 when we add an individual whose best-guess genotype is 0/0 (i.e. the genotype likelihood vector of that individual has the highest value at genotype 0/0), whereas the mode typically moves to the right when we add an individual whose best-guess genotype is 0/1 or 1/1 (Fig. 3B). If we add an individual whose best-guess genotype is 0/1, the mode tends to move one bin to the right and the best-guess allele count increases by 1. By the same token, if we add an individual whose best-guess genotype is 1/1, the mode moves two bins to the right and the best-guess allele count increases by two.

Based on these observations, we propose a new algorithm, called a banded DP algorithm, which can compute the SAF likelihood in a
more efficient manner than the original DP algorithm, but with good approximation to the SAF likelihood computed by the original DP algorithm (Fig. 3C). When updating the SAF likelihood (i.e. filling in each row of the lower triangular matrix), this algorithm finds the most likely position for the mode, and only updates values of the SAF likelihood $d$ bins away from that position in both directions and sets the rest of the values zero. Here $d$ represents a user defined bandwidth.

Let $GC_j$ denote the number of derived alleles in the best-guess genotype for individual $j$ ($GC_j \in \{0, 1, 2\}$) and $AC_j$ the best-guess allele count for $j$ individuals. For each row of the lower triangular matrix, the banded DP algorithm computes the number of derived allele in the best-guess genotype ($GC_j$) based on the genotype likelihoods of a given individual, and then updates the best-guess allele count ($AC_j$) using the best-guess genotype:

$$AC_j = AC_{j-1} + GC_j,$$

where $AC \in [0, 2]$. Next, it computes the values of the SAF likelihood only within $d$ bins away from the best-guess allele count ($AC_j$) in both directions and outside of which it sets the values of the SAF likelihood to zero—i.e. $h_j$ is computed by Equation (5) if $\max (0, AC_j - d) \leq x \leq \min (AC_j + d, 2)$ and otherwise $h_j = 0$. By doing so, the SAF likelihood is updated in a banded-fashion (computing at most $2d + 1$ values at each updating step, where $d$ is a uniform bandwidth) rather than updated in a triangular fashion (computing $3 + 3 + \cdots + (2n + 1)$ values). This makes computation time close to $O(dn)$ rather than the original $O(n^2)$.

Our algorithm has connections with existing banded DP algorithms (for example, banded Needleman-Wunsch alignment algorithm and banded Smith-Waterman alignment algorithm) (Sung, 2009). The banded DP algorithm fills in only the middle part of the DP matrix in a banded fashion (with band length of $2d + 1$), and does not compute the lower and upper triangles in the matrix. However, unlike existing banded DP algorithms, we need to repeat the same process across all sites and we realized that using the uniform value of band length $d$ across all sites does not work in practice due to site-to-site variability of the SAF likelihood (see Fig. 4B). Moreover, around the best-guess allele count, the number of values we need to compute is not usually symmetric in the left and right sides, implying that using the same band length for both sides is not efficient. Therefore, we propose another new algorithm, called a score-limited DP algorithm. This algorithm is different from the banded DP algorithm in a way that the number of cells to be computed at each updating step is adaptively changed rather than being kept at a uniform band length ($2d + 1$).

Our score-limited DP algorithm also starts from computing the number of derived allele in the best-guess genotype based on the genotype likelihoods of a given individual. Then, it proposes left and right boundaries within which we update values of the SAF likelihood. Let $L_j$ and $R_j$ denote the left and right boundaries, respectively, within which we update the SAF likelihood for $j$ individuals ($h_j$). It updates the left and right boundaries using the best-guess genotype:

$$L_j = L_{j-1} + GC_j \quad \text{and} \quad R_j = R_{j-1} + GC_j.$$

For example, for the individuals whose best-guess genotype is 0/0 we do not change the boundaries ($L_j = L_{j-1}, R_j = R_{j-1}$). For individuals whose best-guess genotype is 0/1 we move both boundaries one bin to the right ($L_j = L_{j-1} + 1, R_j = R_{j-1} + 1$), and for the
individuals whose best-guess genotype is 1/1 we move both boundaries two bins to the right ($L_j = L_{j-1} + 2, R_j = R_{j-1} + 2$).

Next, it checks whether a value at the left boundary ($b_{Lj}^i$) is greater than a very small value $\epsilon$ (for example, we set $\epsilon = 10^{-9}$) and if so, it expands the appropriate boundary to the left until the value at the updated left boundary is less than or equal to $\epsilon$. By the same token, it checks the value at the right boundary ($b_{Rj}^i$) and if the value is greater than $\epsilon$, then it expands the right boundary to the right until the value at the updated right boundary is less than or equal to $\epsilon$. Finally, it computes the values of the SAF likelihood only within the left and right boundaries and outside of which we set the values of the SAF likelihood to zero—i.e. $b_{Lj}^i$ is computed by Equation (5) if $\max (0, L_j) \leq x \leq \min (L_j, 2\epsilon)$ and otherwise $b_{Lj}^i = 0$. By doing this at each step of calculating the SAF likelihood, it only computes $R_j - L_j + 1$ number of elements, which is dynamically changing at each updating step, but always much smaller than $2j + 1$. We present the pseudo-code for the score-limited DP algorithm in the Supplementary Appendix.

This revised algorithm has connections with existing score-limited DP algorithms. For example, in a score-limited Smith–Waterman algorithm, the DP matrix is explored in both directions starting from the mid-point of the hit. When the alignment score drops off by more than $\epsilon$ (a user-defined parameter), the extension is truncated. Our adaptive algorithm is similar to this algorithm in that it uses $\epsilon$ to stop computing values of the updated SAF likelihood. We found in practice that our choice of $\epsilon (10^{-9})$ behaves well, but users are encouraged to test the algorithm and decide the appropriate value for themselves ($\epsilon$ is a command-line parameter of our software). Our algorithm is, however, different in that it does not extend from the mid-point in both directions, but it proposed the left and right boundaries and computation is done from right to left. This is required because in real implementation, the updated SAF likelihood for $j$ individuals stored in the same vector for the SAF likelihood for $j - 1$ individuals and in order to properly update values, computation should be done from right to left.

Figure 3 shows with an example how that the score-limited DP algorithm captures the important regions of the SAF likelihood. Hence, the score-limited DP algorithm is faster than the original DP algorithm, as reflected by the reduced computation area (shown in gray in Fig. 3). Moreover, we retain the accuracy of the final SAF likelihood $h$ with the score-limited DP algorithm and it is as stable as the rescaled DP algorithm. The shape of the distribution $h$ is identical in all four cases (original, rescaled, banded and score-limited), reflected by the same mean and variance of $h$ in all four cases (Fig. 3).

3 Methods

3.1 Generating simulated sequences

To compare the four algorithms (original, rescaled, banded and score-limited DP algorithm) for computing SAF likelihoods, we generated aligned short-read sequencing data by changing sequencing coverage ($3 \times$, $5 \times$ and $10 \times$) and sample size (50, 100, 300, 500 and 1000 diploid individuals). For this purpose, we first conducted population genetic simulations to produce haplotype data of a given sample size assuming the standard model (with an effective population size of 10 000 diploid individuals, a mutation rate per-base per-generation of $2.5 \times 10^{-8}$ and a recombination rate of $10^{-8}$), and then overlaid sequencing errors (with error rate of 0.001) to generate paired-end short-read sequencing data given sequencing coverage. For detailed descriptions of the coalescent and sequencing simulations, refer Material and Methods section in Han et al. (2014).

3.2 Sequencing data from the 1000 Genomes Project

To demonstrate the score-limited DP algorithm’s utility with real data, we downloaded the VCF file and the BAM files from the 1000 Genomes Project FTP site in order to estimate the SFS. We used the genotype calls of 365 European and 228 sub-Saharan African individuals from the VCF file, which contains the genotype calls for 1092 individuals sampled from 14 populations drawn from Europe, East Asia, sub-Saharan Africa and the Americas (1000 Genomes Project Consortium, 2010, 2012). For the BAM files, we only used low-coverage Illumina sequencing data (coverage 2× to 4×) (1000 Genomes Project Consortium, 2010, 2012) for these same individuals. Due to file size constraints, we downloaded only a subsection of the genome (region of 10–20 Mb in chromosome 10) by using SAMtools (version 0.1.18) (Li et al., 2009).

3.3 Estimating the SFS

To infer the SFS from simulated aligned short-read sequencing data, we used the direct estimation approach using the freely available program ANGSD (version 0.5.88) with the EM algorithm option to obtain the MLE of the SFS (Nelson et al., 2012). We refer to results of this procedure as Original. Then, we modified the source code of ANGSD to implement the rescaled DP and the score-limited DP algorithm. All code is written in C++.

For the 1000 Genomes Project data, we evaluated two approaches to infer the SFS: the call-based and direct estimation approaches. For the call-based estimation approach, we used genotype calls in the VCF file and then reconstructed the SFS by allele counting using vcTools (version 0.1.10) (Danecek et al., 2011). For the direct estimation approach, we directly estimated the SFS from the BAM files with the score-limited DP algorithm.

To evaluate the accuracy of the SFS estimated from simulated short-read sequencing data, we computed the relative deviation of the inferred SFS (computed from sequencing data) compared with the ground-truth SFS (computed from the known values for the genotype data) in each derived allele frequency bin $i/(2n)$:

$$\text{Relative deviation} = \frac{f_{\text{seq}}(\frac{i}{2n}) - f_{\text{true}}(\frac{i}{2n})}{f_{\text{true}}(\frac{i}{2n})},$$

where $f_{\text{seq}}(\frac{i}{2n})$ represents a fraction of sites with a derived allele frequency $i/(2n)$ in the inferred SFS and $f_{\text{true}}(\frac{i}{2n})$ represents a fraction of sites with a derived allele frequency $i/(2n)$ in the ground-truth SFS.

4 Results

4.1 Performance for changing sequencing coverage

First, we investigated the impact of different sequencing coverage on the performance of the score-limited DP algorithm in computing the SAF likelihood. For this purpose, we simulated sequencing data for
Fast and accurate SFS estimation

50 diploid individuals under the standard model at coverage 3×, 5× and 10×. Figure 4A shows how the SAF likelihood is updated at a random site with true allele frequency of 0.51 as a function of sequencing coverage. We observed that the SAF likelihood h is more diffuse as coverage decreases, whereas it is more peaked around the true allele frequency of 0.51 as coverage increases (Fig. 4A), the variance of the SAF likelihood h is 7.2 with 3×, 2.3 with 5× and 0.21 with 10×). This is because the genotype likelihood vectors tend to be more spread out at low coverage, whereas they tend to be more peaked at the unknown individual genotype at high coverage. This implies that if the banded DP algorithm is used, the choice of bandwidth should depend on coverage—the higher coverage, the smaller d. With this simulated data, we used d of 15 for 3×, 10 for 5× and 5 for 10×, and this worked well for all sites (Fig. 4A, Banded). However, this requires for users to calibrate an appropriate bandwidth before using the algorithm. This difficulty of choosing the appropriate bandwidth is solved with the score-limited DP algorithm. We observed that the optimal bandwidth is adaptively chosen at each updating step (each row of the lower triangular matrix) with the score-limited DP algorithm, and this resulted in a more tight computation area to approximate the SAF likelihood compared with that with the banded DP algorithm. Furthermore, the resulting SAF likelihood has a comparable accuracy to the SAF likelihood computed by the rescaled DP algorithm across all coverage (Fig. 4A). The shape of the distribution h is the same, with the same mean and standard deviation of h, for all three algorithms across coverage.

4.2 Performance for variation in the site likelihood vector

Next, we evaluated that whether the score-limited DP algorithm works well with real data. For this purpose, we used low-coverage sequencing data for 50 diploid GBR individuals in the 1000 Genome Project, and then compared the SAF likelihood computed by the three algorithms (rescaled, banded and score-limited) at multiple random sites with the same best-guess allele frequency in the sample. Compared with the simulated sequencing data matched at average coverage (5×), we observed that low-coverage sequencing data in the 1000 Genomes Project tend to have bigger site-to-site variation of the SAF likelihood. Figure 4B shows how the SAF likelihood h is updated at two random sites with the best-guess allele frequency of 0.5 in the sample. We observed that the first site (position 10 012 499 in chromosome 10) has a bigger variance in the SAF likelihood than the second site (position 10 085 321 in chromosome 10)—the variance of the SAF likelihood at the first site is 12, whereas at the second site is 1.4 (Fig. 4B). Due to large differences in the variability of the SAF likelihoods, the banded DP algorithm with the uniform bandwidth (d = 10) performs badly for the first site but performs well for the second site. However, unlike the banded DP algorithm, the score-limited DP algorithm performs well for both sites, because it is capable of changing the bandwidth accordingly to the observed variance of the SAF likelihood at different sites—the larger the variance of the SAF likelihood, the larger the bandwidth. Moreover, the resulting SAF likelihood computed by the score-limited DP algorithm has a comparable accuracy to that computed by the rescaled DP algorithm—same shape, and the mean and standard deviation with the three algorithms (Fig. 4B).

4.3 Evaluating the accuracy of the inferred SFS

We evaluate the accuracy of the inferred SFS by the score-limited DP algorithm (score-limited) compared with the inferred SFS using the original DP (original). For this comparison, we simulated 100 replicates of sequencing data for 100, 300 and 500 diploid individuals each from genomic regions of length 100 Kb under the standard model. The accuracy of the inferred SFS was evaluated by two metrics: (i) the shape of the inferred SFS in comparison to the ground-truth SFS (Fig. 5A) and (ii) the relative deviation of the inferred SFS compared with the ground-truth SFS at each allele frequency bin (Fig. 5B).

We found that the score-limited DP algorithm behaves equivalently to the original DP algorithm. We observed the identical shape of the inferred SFS (Fig. 5) with both algorithms. Moreover, consistent with our previous study (Han et al., 2014), both algorithms led to unbiased estimates of the SFS even at low coverage (such as 3×) regardless of sample size. The shape of the inferred SFS was similar to the ground-truth SFS (Fig. 5A) and the relative deviation of the inferred SFS was close to 0 in all allele frequency bins (Fig. 5B) across all sequencing coverage.

4.4 Runtime comparisons

We next evaluated the runtime for computing site likelihood vectors by the score-limited DP algorithm (score-limited) compared with the runtime by the original DP algorithm (original). We observed runtime speed-ups with the score-limited DP algorithm compared with the original DP algorithm for all sample sizes we tested (Fig. 1). For example, on average, with 500 individuals we observed 5.3-fold speed-up, and with 1000 individuals we observed 8.4-fold speed up. Moreover, consistent with our expectation, the runtime of the original DP algorithm increases quadratically with sample size, whereas the runtime of the score-limited DP algorithm has a linear fit (Fig. 1). These results imply that as a sample size increases, we will observe even more dramatic differences in the runtimes of the two algorithms. For example, when we extrapolated runtime from the results with 2000 individuals, we expect 17.5-fold fold-up with 2500 individuals and 63-fold fold-up with 10 000 individuals. The speed improvement with the score-limited DP algorithm will greatly facilitate direct inference of the SFS even when the number of individuals is large.

We want to emphasize that the runtime in Figure 1 is per 10 Mb region, and for the whole genome the runtime would be greater by a factor of 300. However, computational efficiency can be further improved by distributing SAF likelihood computation across nodes. Another possibility is sub-sampling random sites across the genome and distributing the SAF likelihood computation across nodes based on only those sub-sampled sites.

We also note that the score-limited DP algorithm will have less memory usage than the original DP algorithm, which requires memory on the order of n to store the SAF likelihood. With the score-limited DP algorithm, the memory needed is to store the Rn – Ln + 1 elements of the vector, and we expect that number to stay nearly constant or to scale upwards slowly in proportion to n. In our implementation, when writing the output of the SAF likelihood to a file, at each site, we only output non-zero values of the SAF likelihood, the left boundary (L), and the number of non-zero values (Rn – L + 1), rather than outputting all 2n + 1 values. This vastly saves file size to store SAF likelihoods across all sites, and requires less memory to read in the SAF likelihood file to subsequently run an EM algorithm or a BFGS algorithm. For example, for simulated sequencing data of 500 individuals at 5×, the size of the SAF likelihood file is 764 Mb with the original DP algorithm, whereas that with the score-limited DP algorithm is 8.9 Mb. For simulated sequencing data of 1000 individuals at 5×, the file size with the
morphic sites in the genetic region we analysed than the reported polymorphic sites in the VCF file provided from the 1000 Genome Project (data not shown).

Next, we inferred the SFS for 228 AFR individuals and a combined sample of 593 EUR and AFR individuals with either the call-based approach or the direct estimation approach. We observed a similar pattern as with the European population, implying that our results apply to all samples in the 1000 Genomes Project (data not shown).

5 Discussion
A large sample size enables us to infer more precise summary statistics and parameters in many population genetic analyses. However, at the same time, we confront computational challenges with large samples and in many cases, we have to deal with these challenges to make the method practical with large sample sizes. We showed that although the direct estimation approach for computing the SFS can provide the unbiased SFS even at low coverage, it does not scale up to large sample sizes because the computation time for running this method is quadratic in a number of diploid individuals. To overcome this problem, we developed a new algorithm, called the score-limited DP algorithm, and showed that the computation time for running this algorithm is linear in the number of genomes. This algorithm exploits the observation that for most sites the SAF likelihood’s non-negligible values are all concentrated on a few elements around the element corresponding to the best-guess allele count. Therefore, we approximate this vector by curtailing computation to only a few components of the DP update vectors. More importantly,
this algorithm can adaptively choose the bandwidth \( d \) during updating the SAF likelihood for each site. We showed that the bandwidth change is robust to sequencing coverage and the variation of the SAF likelihood. We also showed that the EM combined this new algorithm has comparable accuracy but is \( b \)-fold faster than the original DP combined with the EM algorithm when analysing the data from 1000 individuals. Our new algorithm's improvement in speed makes it possible to directly estimate the SFS from very large samples of low coverage short-read sequencing data.

Our score-limited DP algorithm could be applied to other DP algorithm whose runtime is quadratic in a sample size. For example, Yi et al. (2010) proposed an empirical Bayes approach to estimate a posterior probability of a minor allele frequency (MAF). They used a DP algorithm to effectively compute summation over all possible genotype configurations for \( n \) diploid individuals, and therefore this algorithm has a runtime complexity of \( O(n^2) \) similar to the DP algorithm introduced here. Furthermore, similar to the distribution of the SAF likelihood, the distribution of the posterior probabilities of the MAF is unimodal and most of the probabilities are close to 0. Therefore, we can apply our score-limited DP algorithm for this DP algorithm to reduce runtime complexity to be \( O(dn) \) rather than original \( O(n^2) \) where \( d \) is the maximum bandwidth.

Our score-limited DP algorithm can also be directly applied to speed up estimation of the 2D SFS. Li (2011) derived the EM algorithm to get the MLE of the 2D SFS as an extension to the 1D SFS estimation, and this requires precomputation of the SAF likelihoods for all sites for each population independently. This implies that we can make this method faster with the score-limited DP algorithm compared with the original DP algorithm. The computation time for running the original algorithm is \( O(n_1^2 + n_2^2) \), whereas the runtime of the score-limited DP algorithm becomes \( O(d_1 n_1 + d_2 n_2) \), where \( n_1, n_2 \) represent a sample size for each population and \( d_1, d_2 \) are the maximum bandwidth.

One might argue that uncertainty associated with genotype calls can be overcome by simply increasing sequencing coverage and there is therefore little need for algorithms that handle low coverage data. However, cost constraints require different choices between increasing sample size and increasing coverage. There are certain cases where one prefers a large sample of low-coverage sequencing data over a smaller sample size with high coverage. For example, in genome-wide association studies, one can obtain more power by sequencing a large number of individuals at low coverage (Kim et al., 2010; Pasaniuc et al., 2012). As another example, identification of rare variants always requires large sample sizes, and moderately rare loci will be detectable even with low coverage data. Finally, even though sequencing cost keeps dropping, cost constraints will not disappear because users will continue to work with limited budgets and push these limits with applications involving very large numbers of individuals; thus we expect low-coverage sequencing will remain an attractive approach for many investigators and that methods like ours will retain their appeal for the foreseeable future.

**Acknowledgement**

We thank Darren Kessner for his assistance with the sequencing simulations.

**Funding**

This study was funded by National Institutes of Health [T32 HG002536 to E.H., GM033275 to J.S. and HG007089 to J.N.].

**Conflict of interest:** none declared.

**References**


