BIOINFORMATICS ORIGINAL PAPER Vol. 24 no. 14 2008, pages 1596–1602
doi:10.1093/bioinformatics/btn236

Genetics and population analysis

Penalized estimation of haplotype frequencies

Kristin L. Ayers[1,*] and Kenneth Lange[1,2,3,*]

[1]Department of Biomathematics, [2]Department of Human Genetics and [3]Department of Statistics, University of California, Los Angeles, CA 90095, USA

Received on March 29, 2008; revised on May 13, 2008; accepted on May 14, 2008

Advance Access publication May 16, 2008

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Low haplotype diversity and linkage disequilibrium are the rule in short genomic segments. This fact suggests that parsimony should be enforced in estimation of haplotype frequencies. The current article introduces a diversity penalty that automatically discards potential haplotypes with low explanatory power. The standard EM algorithm for haplotype frequency estimation can accommodate the penalty if one passes over to a more general minorize–maximize (MM) scheme for estimation.

Results: Our new MM algorithm converges in fewer iterations, eliminates marginal haplotypes from further consideration and reduces the computational complexity of each iteration. Estimation by the MM algorithm also improves haplotyping and genotype imputation compared to naive application of the EM algorithm. Thus, the MM algorithm is a useful substitute for the EM algorithm. Compared to the most sophisticated current methods of haplotyping and genotype imputation, the MM algorithm is slightly less accurate but at least an order of magnitude faster.

Availability: Our software will be made available in the next release the program Mendel at http://www.genetics.ucla.edu/software/.

Contact: kayers@ucla.edu

1 INTRODUCTION

Estimation of haplotype frequencies serves a variety of purposes. For example, good estimates help distinguish ethnic groups, quantify the extent of linkage disequilibrium and guide imputation of missing genotypes. In mapping Mendelian disease genes, haplotype signatures provide evidence of unique mutation events. In association studies with common diseases, these signatures can offer more definitive predictors than single marker alleles (Akey *et al.*, 2001; Ayers *et al.*, 2007). With the advent of large-scale genome association studies and massive single nucleotide polymorphism (SNP) genotyping, haplotyping and associated tasks have taken on greater urgency. Fortunately, the enormous energy expended by geneticists in improving haplotyping is beginning to pay dividends in faster and more accurate software. Halperin and Eskin (2004), Scheet and Stephens (2006) and Marchini *et al.* (2006) summarize and compare the recent computational approaches.

The EM algorithm lying at the heart of many of these methods relies on a classical gene counting argument (Excoffier and Slatkin, 1995; Hawley and Kidd, 1995; Long *et al.*, 1995; Qin *et al.*, 2002). The algorithm operates on population data by filling in missing phase information based on current haplotype frequencies. Given reconstructed phases, the EM algorithm equates haplotype frequencies to imputed haplotype proportions. This iterative process of imputation and re-estimation is natural and effective. One of its strengths is that it accommodates a Dirichlet prior on haplotype frequencies (Lange, 2002). In this Bayesian context, the EM algorithm simply adds fixed pseudo-counts to imputed counts before forming its new haplotype proportions. The drawback of a Dirichlet prior is that it can only encourage the inclusion of rare haplotypes. If we want to discourage the inclusion of rare haplotypes with low explanatory power, we must turn elsewhere. In this article we propose a haplotype diversity penalty that has the desired opposite effect. Simple modification of the EM algorithm yields a novel algorithm that maximizes the penalized likelihood.

Our algorithm is example of an minorize–maximize (MM) algorithm. All EM algorithms are MM algorithms but not vice versa. Many MM algorithms, ours included, dispense with the missing data structures required by EM algorithms. In their stead one must construct a surrogate function that is optimized at each iteration. Derivation of surrogate functions requires manipulation of mathematical inequalities. Compared to the traditional EM algorithm for haplotype frequency estimation, our new MM algorithm converges in fewer iterations, eliminates marginal haplotypes from further consideration and reduces the computational complexity of each iteration. Imposition of the diversity penalty also improves haplotyping and genotype imputation. Compared to more sophisticated methods of haplotyping such as PHASE (Marchini *et al.*, 2006; Stephens *et al.*, 2001) and fastPHASE (Scheet and Stephens, 2006; Stephens and Scheet, 2005), the MM algorithm is slightly less accurate but considerably faster.

2 METHODS

An MM algorithm for maximization involves minorizing an objective function $f(p)$ by a surrogate function $g(p\,|\,p^n)$ anchored at the current iterate p^n of a search (De Leeuw and Heiser, 1977; Groenen, 1993; Hunter and Lange, 2004; Lange, 2004). Minorization is defined by the two properties

$$f(p^n) = g(p^n\,|\,p^n) \tag{1}$$

$$f(p) \geq g(p\,|\,p^n), \quad p \neq p^n. \tag{2}$$

In other words, the surface $p \mapsto g(p\,|\,p^n)$ lies below the surface $p \mapsto f(p)$ and is tangent to it at the point $p = p^n$. Construction of the minorizing function $g(p\,|\,p^n)$ constitutes the first M of the MM algorithm.

The second M of the algorithm maximizes the surrogate $g(p\,|\,p^n)$ rather than $f(p)$. If p^{n+1} denotes the maximizer of $g(p\,|\,p^n)$, then this action forces the ascent property $f(p^{n+1}) \geq f(p^n)$. The proof of the property follows from the inequalities

$$f(p^{n+1}) \geq g(p^{n+1}\,|\,p^n) \geq g(p^n\,|\,p^n) = f(p^n)$$

*To whom correspondence should be addressed.

reflecting the definition of p^{n+1} and the tangency conditions (1) and (2). The ascent property lends the MM algorithm great numerical stability. Because minorization is closed under the formation of sums, many objective functions can be minorized piece by piece.

It is instructive to derive the traditional EM algorithm for haplotype frequency estimation from the MM perspective. Let H_i be the set of maternal–paternal haplotype pairs consistent with the observed genotype of person i at each marker. If p_j is the frequency of haplotype j, then the likelihood of i's observed multi-marker genotype is

$$r_i = \sum_{(k,l)\in H_i} p_k p_l.$$

Our MM derivation exploits the concavity of the function $\ln x$ and minorizes the loglikelihood $L(p)$ of the whole sample by

$$\begin{aligned} L(p) &= \sum_i \ln r_i \\ &\geq \sum_i \sum_{(k,l)\in H_i} \frac{p_k^n p_l^n}{r_i^n} \ln\left(\frac{r_i^n}{p_k^n p_l^n} p_k p_l\right) \\ &= \sum_j c_j^n \ln p_j + c_0^n, \end{aligned}$$

where c_j^n is a positive constant that depends on the previous parameter vector p^n but not on the current parameter vector p, and j ranges over all haplotypes consistent with at least one multi-marker genotype. A brief calculation shows that

$$\begin{aligned} c_j^n &= \sum_i \sum_{(k,l)\in H_i} \left(1_{\{k=j\}} + 1_{\{l=j\}}\right)\left(\frac{p_k^n p_l^n}{r_i^n}\right) \\ c_0^n &= \sum_i \sum_{(k,l)\in H_i} \frac{p_k^n p_l^n}{r_i^n} \ln\left(\frac{r_i^n}{p_k^n p_l^n}\right). \end{aligned}$$

The M step of the EM algorithm maximizes the surrogate function $\sum_j c_j^n \ln p_j + c_0^n$ subject to the linear equality constraint $\sum_j p_j = 1$ and the lower bounds $p_j \geq 0$. Note that the surrogate function separates parameters. This desirable feature carries over to the penalized loglikelihood.

Our diversity penalty is modeled on the lasso penalty, which was introduced in regression analysis to perform continuous model selection and enforce sparse solutions in underdetermined problems (Chen *et al.*, 1998; Claerbout and Muir, 1973; Santosa and Symes, 1986; Taylor *et al.*, 1979; Tibshirani, 1996). Unfortunately, the lasso penalty $\lambda \sum_j |p_j| = \lambda$ is worthless in the haplotype setting because it simply reduces to the tuning parameter λ. A more sensible penalty is linear for small haplotype frequencies and levels off thereafter. We therefore suggest the penalty $\lambda \sum_j f(p_j)$, where

$$f(q) = \begin{cases} q & q \leq \delta \\ \delta & q \geq \delta \end{cases}$$

for some positive threshold δ. This choice of the penalty still discourages small positive estimates. The optimal value of the tuning constant λ can be determined by numerical experimentation.

The overall minorization

$$L(p) - \lambda \sum_j f(p_j) \geq \sum_j c_j^n \ln p_j + c_0^n - \lambda \sum_j f(p_j).$$

now involves non-differentiable penalty terms. To handle these, we majorize the penalty function $f(q)$ by smooth functions. There are two cases to consider. When $q^n < \delta$, we minorize $-f(q)$ by $-q$. When $q^n \geq \delta$, we minorize $-f(q)$ by the constant $-\delta$. One can easily draw a simple graph illustrating how the tangency conditions are met in each case; see Figure 1. If S^n is the haplotype set $\{j : p_j^n < \delta\}$, then the surrogate function minorizing the penalized loglikelihood is

$$\sum_j c_j^n \ln p_j + c_0^n - \lambda \sum_{j\in S^n} p_j - \lambda \sum_{j\notin S^n} \delta.$$

Parameters continue to be separated.

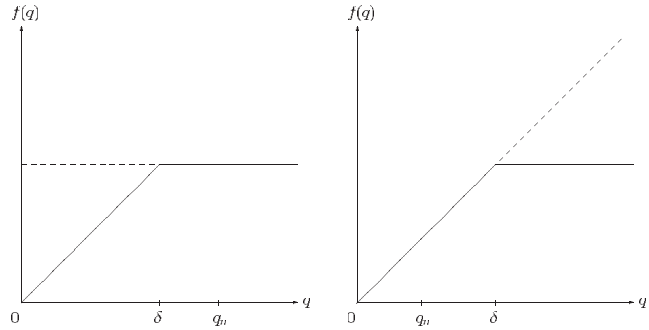

Fig. 1. The penalty function $f(q)$ and its majorizer $g(q|q_n)$. $f(q)$ is plotted as a bold line and $g(q|q_n)$ as a dashed line.

In maximizing the surrogate function, the bound $p_j \geq 0$ can be ignored because the term $c_j^n \ln p_j$ tends to ∞ as p_j tends to 0. The equality constraint $\sum_j p_j = 1$ must be faced, however. This is done by introducing a Lagrange multiplier ω and looking for a stationary point of the Lagrangian

$$\begin{aligned} \mathcal{L}(p) = &\sum_j c_j^n \ln p_j + c_0^n - \lambda \sum_{j\in S^n} p_j \\ &- \lambda \sum_{j\notin S^n} \delta + \omega\left(\sum_j p_j - 1\right). \end{aligned}$$

Thus, we must solve the equations

$$\frac{\partial}{\partial p_j} \mathcal{L}(p) = \frac{c_j^n}{p_j} - \lambda 1_{S^n}(j) + \omega = 0. \tag{3}$$

If we multiply Equation (3) by p_j and sum on the index j, then the constraint $\sum_j p_j = 1$ requires

$$\omega = -\sum_j c_j^n + \lambda \sum_{j\in S^n} p_j = -\sum_j c_j^n + \lambda t,$$

where $t = \sum_{k\in S^n} p_k$. Substituting this result in Equation (3) produces

$$p_j = \begin{cases} \frac{c_j^n}{\sum_k c_k^n - \lambda t} & j \notin S^n \\ \frac{c_j^n}{\sum_k c_k^n - \lambda t + \lambda} & j \in S^n. \end{cases} \tag{4}$$

For these solutions to be consistent with the constraint, we must have

$$1 = \frac{1}{\sum_k c_k^n - \lambda t} \sum_{j\notin S^n} c_j^n + \frac{1}{\sum_k c_k^n - \lambda t + \lambda} \sum_{j\in S^n} c_j^n.$$

If we let $d = \sum_{j\notin S^n} c_j^n$ and $e = \sum_{j\in S^n} c_j^n$, then we can recast this condition as

$$1 = \frac{1}{d + e - \lambda t} d + \frac{1}{d + e - \lambda t + \lambda} e.$$

In the exceptional cases $d = 0$ and $e = 0$, the values $t = 1$ and $t = 0$ clearly work. In both cases the MM update reduces to the EM update. Otherwise, cross multiplying by $(d + e - \lambda t)(d + e - \lambda t + \lambda)$ and rearranging terms leads to the quadratic

$$\lambda t^2 - (d + e + \lambda)t + e = 0, \tag{5}$$

with solution

$$t = \frac{(d + e + \lambda) - \sqrt{(d + e + \lambda)^2 - 4\lambda e}}{2\lambda}.$$

Because the quadratic on the left-hand side of Equation (5) has value e at the point $t = 0$ and value $-d$ at the point $t = 1$, it is clear that its smaller root is the pertinent one. Furthermore, the smaller root lies on the open interval (0,1). Substituting the smaller root for t in formula (4) fully specifies the MM update. It is clear from this exercise that the MM algorithm retains most of

the computational simplicity of the EM algorithm for haplotype frequency estimation. No matrix operations are required, and penalization is built in.

Our computer implementation of the MM algorithm in the genetic analysis program Mendel (Lange *et al.*, 2001) simultaneously conducts haplotype frequency estimation, haplotyping and genotype imputation. Mendel uses a haplotype window surrounding a central marker flanked by f markers on the left and f markers on the right. The central marker is the object of phase and genotype imputation. The value of f is determined by the user. Mendel's current default of 9 gives a window of length 19. Estimation of haplotype frequencies commences with a defined list of haplotypes much shorter than the full list of available haplotypes. We will comment in a moment on how this list is generated and how windows at the ends of chromosomes are handled. If an individual is untyped at a marker, then during haplotype frequency estimation, all genotypes at the marker are assumed possible for the individual. Mendel takes initial haplotype frequencies to be uniform and iterates via the MM algorithm until the ℓ_1 distance $\sum_k |p_k^{n+1} - p_k^n|$ between successive iterations n and $n+1$ drops below 10^{-4} or the number of iteration exceeds 100. Once convergence is declared, Mendel discards all haplotypes with estimated frequencies below 10^{-8}.

Given haplotype frequency estimates, Mendel imputes phase at the central marker for a given person by finding the ordered genotype at the central marker with the highest posterior probability over all consistent haplotype pairs. In the absence of pedigree data, this discovery by itself does not pin phase down. However, if we imagine sliding the haplotype window from left to right across a chromosome, then imputed ordered genotypes to the left of the central marker will be available. We can therefore assign phase to any consistent haplotype pair. If a consistent haplotype pair that agrees with the already imputed phases is not found, Mendel will search for haplotypes pairs that disagree at only one position. To the left of the central marker, a mismatch can involve either one phase switch or one allele mismatch. Allele mismatches are not allowed at the current central marker. To the right of the central marker, a mismatch can involve only allele mismatches because phase has not yet been imputed. In the rare case that a consistent haplotype pair is still not found, the most common genotype is used to fill in the missing ordered genotype.

Imputation of missing genotypes in the absence of haplotyping is handled a little differently. We now divide the consistent haplotype pairs into groups depending on the unordered genotype at the central marker. Mendel assigns a probability to each group by summing the product probabilities of its haplotype pairs. If no consistent haplotype pairs are found, then Mendel will allow for one allele mismatch. The group with highest probability determines the missing genotype at the central marker. In other words, Mendel selects the unordered genotype with highest posterior probability.

When we slide the haplotype window one marker to the right, we must construct a new abbreviated list of possible haplotypes. As we mentioned, we discard haplotypes from the existing list with estimated frequencies below 10^{-8}. For each remaining haplotype, we crop its leftmost allele and add on its right one of the possible alleles at the new marker. If the new marker has m alleles, this action propagates each cropped haplotype into m different haplotypes in the new list. For example, the current SNP haplotype 1-2-2-1-2 is cropped to 2-2-1-2 and expanded to the two new haplotypes 2-2-1-2-1 and 2-2-1-2-2. Our retention–propagation strategy keeps all pertinent haplotypes in play. Penalization weeds outs many of the haplotypes in the new list and keeps the list from growing geometrically.

This description omits initialization of the haplotype list. Since computation times scale as the square of list length, it is imperative to adopt a strategy that minimizes list length. Thus, at the leftmost marker, we start with a window of length 1 and extend it as just described, except for imputation and cropping, until it hits length $f+1$. At that point, we commence haplotype and genotype imputation at the leftmost marker but still omit cropping. When the window reaches full length $2f+1$, then we begin haplotype cropping. At the right end of the chromosome, haplotype propagation is omitted as soon as new markers are exhausted. Haplotype and genotype imputation continue until the rightmost marker is processed. These tactical adjustments entail more book keeping and shift the focus away from the center of the window. In compensation, they successfully keep all haplotype lists short.

In some regions little or no linkage disequilibrium exists, and the number of haplotypes can balloon out of control. Many individuals will have unique haplotypes; other individuals will be consistent with many haplotype pairs. The result is a large list of haplotypes, with many haplotypes having equal frequency estimates. In these regions, genotype imputation is already poor. To decrease computation time, we limit the number of haplotypes in a window to $h_{\max}$. We order the haplotypes by decreasing frequency and find the frequency of haplotype $h_{\max}+1$. Any haplotype with frequency less than or equal to this amount is dropped before moving to the next window. In very rare cases, this tactic deletes too many haplotypes, so we impose a lower limit $h_{\min}$ on the number of haplotypes retained. When $h_{\max} = h_{\min}$, all $h_{\max}$ haplotypes are kept. Setting a rigorous bound on retained haplotypes is also central to other haplotyping programs such as SNPHAP (http://www-gene.cimr.cam.ac.uk/clayton/software/).

3 RESULTS

3.1 Haplotype frequency estimation

To compare the MM and the EM algorithms in haplotype frequency estimation, we randomly generated multilocus autosomal genotypes from male X chromosome haplotypes. The fathers in the 30 European (CEU) parent–offspring trios of the HapMap project are a convenient source of data (http://www.hapmap.org). We chose groups of fully typed consecutive markers outside the pseudoautosomal region with 8–11 markers per group. We then simulated 100 sets of 50, 100 and 500 genotypes from each group, sampling haplotypes with replacement. For this analysis, we began estimation with the list of all consistent haplotypes. Table 1 gives the results of applying the MM algorithm as a function of δ and λ for 100 genotypes. Results for 50 and 500 genotypes were similar (data not shown). The EM algorithm correspond to the choice $\lambda = 0$. The error column gives the average value of the ℓ_1 error $\sum_i |\hat{p}_i - p_i|$ over all replicates and all marker groups of a given size. Here p_i is the generating haplotype frequency, and $\hat{p}_i$ is the estimated frequency. Average squared error and average maximum error lead to similar conclusions. It is clear from the table that the MM algorithm takes fewer iterations and much less computing time to converge than the EM algorithm. Error rates are modestly better under the MM algorithm. The error surface is relatively flat in λ. As a rule of thumb, we suggest choosing λ between 100 and 1000, with larger values for larger sample sizes. Error rates are also relatively flat in δ. We recommend the choice $\delta = 0.005$ for haplotype frequency estimation in a small window. To achieve the highly accurate estimates in this test, we departed from Mendel's defaults and chose the more stringent ℓ_1 convergence criterion of 10^{-5} and the more liberal value of 150 for the maximum number of iterations.

To test how the EM and MM algorithms perform in conjunction with our specific haplotype extension strategy, we simulated 100 autosomal genotypes using a longer stretch of the same HapMap X chromosome data. All 30 European haplotypes in this region of 110 consecutive markers are distinct. We initiated estimation at the first marker and extended haplotypes by adding one marker at a time. At each extension step, we computed haplotype frequencies and dropped those haplotypes with frequencies below 10^{-8}. At most 100 haplotypes were retained at each step. Table 2 records our average results over 100 random replicates for $\delta = .005$ and $\lambda = 100$ and $\lambda = 1000$. Column 1 gives window length, Column 2 the number of iterations until convergence, Column 3 the time in seconds

Table 1. Deviations of computed haplotype frequencies from their true values for 100 individuals and 4 datasets

	$\delta = 0.001$			$\delta = 0.005$			$\delta = 0.01$		
λ	Iter	ℓ_1 error	Time(s)	Iter	ℓ_1 error	Time(s)	Iter	ℓ_1 error	Time(s)
	8 *markers* (6)								
0	23.72	0.1097	2.3279						
10	22.23	0.1097	2.3914	23.03	0.1095	2.4897	23.03	0.1094	2.4852
100	16.17	0.1096	1.7561	17.23	0.1085	1.9107	18.24	0.1083	2.0274
1000	12.11	0.1085	1.3168	7.55	0.1085	0.8396	11.73	0.1354	1.3276
10000	11.38	0.1085	1.2466	5.88	0.1087	0.6663	10.80	0.1408	1.2520
100000	11.04	0.1085	1.2264	5.73	0.1087	0.6564	10.62	0.1414	1.2367
	9 *markers* (16)								
0	65.88	0.2325	2.5023	62.10					
10	59.60	0.2323	2.7621	47.57	0.2278	2.1710	49.53	0.2271	2.2536
100	42.01	0.2307	1.9465	39.43	0.2244	1.8293	43.59	0.2293	1.9856
1000	33.42	0.2265	1.5160	35.38	0.2671	1.6197	57.05	0.3863	2.7000
10000	33.41	0.2285	1.4856	36.56	0.3125	1.7034	55.19	0.4257	2.6065
100000	33.39	0.2285	1.4640	35.72	0.3148	1.6629	54.18	0.4326	2.5368
	10 *markers* (6)								
0	22.52	0.1160	4.7730						
10	21.59	0.1160	4.6272	22.08	0.1159	4.7307	22.11	0.1158	4.7277
100	15.35	0.1151	3.2193	17.80	0.1141	3.9100	18.18	0.1142	3.9349
1000	9.85	0.1138	2.1119	8.76	0.1196	2.0040	11.52	0.1377	2.6621
10000	8.86	0.1144	1.9443	10.31	0.1527	2.5059	13.45	0.2014	3.2914
100000	8.62	0.1144	1.9076	10.36	0.1535	2.5302	13.37	0.2035	3.3021
	11 *markers* (7)								
0	29.69	0.1251	9.3812						
10	28.76	0.1251	10.2635	27.46	0.1250	9.8979	27.40	0.1250	9.9489
100	19.95	0.1246	7.1080	16.83	0.1241	6.0724	19.16	0.1244	7.2570
1000	15.27	0.1246	5.4492	12.42	0.1256	4.7980	16.71	0.1361	6.5768
10000	16.15	0.1249	5.9050	11.18	0.1259	4.3932	15.49	0.1385	6.4742
100000	15.46	0.1246	5.7591	10.64	0.1259	4.1641	15.18	0.1385	6.3782

Note: The number of generating haplotypes is shown in parentheses next to the number of markers.

until convergence, Column 4 the ℓ_1 error, Column 5 the maximum number of haplotypes encountered and Column 6 the actual number of haplotypes in the sample. Sampling was done with replacement, and time is cumulative. For testing convergence, we used Mendel's default criteria.

Some interesting conclusions emerge from the table. First, standard EM does surprisingly well when paired with our extension–elimination strategy. Nonetheless, MM takes about half as many iterations and about half the time. For this increase in speed, MM pays a small but manageable price in ℓ_1 error. Error rates stabilize because there are only 30 generating haplotypes. Once all of these are included in the model, accuracy remains almost constant as new markers are added.

3.2 Genotype imputation

To compare the performance of the MM and EM algorithms in genotype imputation, we analyzed the X chromosome HapMap data on all 54 males of the African population (Yoruban). We first removed pseudoautosomal markers and markers with missing genotypes. After the data were cleaned, we constructed 30 genotypes by sampling haplotypes with replacement from the first 10 000 markers. We then randomly deleted 1% of the constructed genotypes. These steps generated 3008 missing genotypes and positioned us to evaluate the accuracy of the MM and EM algorithms in genotype imputation. In our experience, imputation by posterior probability is more accurate than imputation by most likely haplotype pair. Table 3, therefore records counts of imputation errors using posterior probabilities. Since error rates depend on the number of flanking markers, the table lists results in the range of 6–10 flanking markers. In the table, C_m denotes the number of incorrectly imputed genotypes, and A_m denotes the number of incorrectly imputed alleles. These numbers differ slightly because a few imputed genotypes incorrectly specify both alleles.

The EM algorithm is overwhelmed by the sheer number of haplotypes when the number of flanking markers reaches eight. The MM algorithm discards most haplotypes and can attack much longer segments. Introducing strict limits on the number of haplotypes within a window allows the EM algorithm to recover. Haplotype frequency estimation was performed under Mendel's default convergence criterion and an upper limit of $h_{\max} = h_{\min} = 100$ haplotypes per window. Compared to more stringent criteria, these choices greatly reduce computing times with virtually no effect on error rates.

Inspection of Table 3 shows that both error rates reach their approximate minima for nine flanking markers and the value $\lambda = 1000$. Recall that $\lambda = 0$ corresponds to the EM algorithm. Experiments not displayed in the table suggests that the choice $\delta = .005$ performs almost as well as our current choice $\delta = .01$. At the bottom of the table, we list the more accurate but far slower results of fastPHASE. For fastPHASE, we invoked the options −H−4

Table 2. Haplotype frequency estimation via marker extension

Window length	Iter	Time(s)	ℓ_1 error	H_e	H_s
EM					
2	12.30	0.0078	0.00000398	4.00	3.00
10	17.07	0.0657	0.01142453	13.16	7.00
30	27.62	1.4685	0.02444192	33.20	14.99
40	36.93	3.5966	0.02777934	49.34	21.99
50	33.42	5.9615	0.02653682	49.08	21.99
60	27.67	8.5778	0.01168201	57.68	25.98
70	22.83	10.9654	0.01034319	60.58	26.98
80	13.03	12.2991	0.00481665	62.34	27.98
90	10.01	13.7276	0.00451555	70.78	28.98
100	8.20	14.8207	0.00436692	70.60	28.98
110	10.61	16.1275	0.00485051	77.56	29.98
MM	λ = 100				
2	10.71	0.0074	0.00000169	4.00	3.00
10	12.01	0.0551	0.01110438	12.72	7.00
20	18.24	0.3906	0.02227146	26.96	14.00
30	18.99	0.9869	0.02176035	32.90	14.99
40	22.28	2.2788	0.02089113	48.16	21.99
50	22.09	3.7472	0.01945703	47.36	21.99
60	19.05	5.3511	0.00715012	55.34	25.98
70	16.68	6.9214	0.00605148	57.20	26.98
80	9.55	7.9332	0.00354929	60.34	27.98
90	7.89	8.9868	0.00340693	68.18	28.98
100	7.28	9.8501	0.00335740	66.36	28.98
110	8.37	10.8586	0.00374288	72.88	29.98
MM	λ = 1000				
2	8.64	0.0073	0.00000022	4.00	3.00
10	9.44	0.0456	0.01208444	12.04	7.00
20	9.05	0.2288	0.02092093	24.68	14.00
30	8.36	0.5181	0.02095165	31.46	14.99
40	9.01	1.0461	0.01949569	46.90	21.99
50	8.40	1.6180	0.01885398	45.52	21.99
60	7.90	2.3104	0.00803921	54.34	25.98
70	7.25	3.0631	0.00567504	56.20	26.98
80	6.11	3.6691	0.00441167	57.76	27.98
90	5.96	4.3387	0.00711512	60.82	28.98
100	6.22	4.9303	0.00805822	59.96	28.98
110	5.86	5.6075	0.00506157	64.66	29.98

and −K10. The −H option shuts off haplotype estimation, and the −K10 options sets the number of haplotype clusters to 10. Both of these choices promote faster computation times at the expense of a slight increase in error rates.

We also compared results on two other populations with 60 individuals each from the SeattleSNPs resequencing project (http://pga.gs.washington.edu). Our initial findings on 50 different genes (data not shown) are similar to the HapMap findings. The SeattleSNPs analysis was also done in both PHASE v2.1 and fastPHASE. The software for these programs were downloaded at http://www.stat.washington.edu/stephens/software.html. We found PHASE to have similar error rates to fastPHASE, varying by a fraction of a percent, but much longer computation times. Because of PHASE's inability to handle large numbers of markers simultaneously, we abandoned PHASE on a comparison involving 10 000 markers. In this larger dataset, the simple default of filling in missing genotypes with the most common genotype in the population results in 873 mistakes for an error rate of 29%. Mendel reduces this error rate to 4.6%, and fastPHASE reduces it further to 2.5%. In timing comparisons Mendel is about 50 times faster than fastPHASE.

3.3 Haplotyping

To compare the MM and EM algorithms on large-scale haplotyping, we reverted to the simulated data constructed from the African HapMap X chromosome data. Again we elected Mendel's default convergence criteria and set $h_{\max} = h_{\min} = 100$. In this case, we filled in missing phases and missing genotypes using the ordered genotypes rather than the unordered genotypes with the highest posterior probabilities. Table 4 records the number C_m of incorrectly imputed genotypes and the number C_s of phase switch errors under this strategy. Markers with missing genotypes were not included in the switch error because an imputed genotype can differ from the true genotype. The bottom line of the table displays fastPHASE's result on the same data under the −K10 option.

In this comparison, Mendel's best genotype imputation error rate of 4.9% is nearly double fastPHASE's error rate of 2.6%. Mendel's best phase switch error rate of 5.4% is also about double fastPHASE's error rate of 2.8%. Increasing the number of flanking markers continues to improve Mendel's phase switch error rate, but it eventually increases the genotype imputation error rate.

Table 3. Genotype imputation errors for a 10K dataset with 30 genotypes

Flanking markers	6			7			8			9			10		
λ	C_m	A_m	Time (s)	C_m	A_m	Time (s)	C_m	A_m	Time (s)	C_m	A_m	Time (s)	C_m	A_m	Time (s)
0	172	174	98.265	158	160	124.302	163	167	163.22	160	163	209.075	170	176	240.040
10	168	170	81.984	163	165	104.895	164	167	136.63	156	159	180.004	168	173	200.036
100	163	166	58.472	152	154	75.754	156	160	95.112	151	156	122.243	157	162	146.226
1000	159	162	40.402	146	150	54.741	142	145	68.310	**137**	143	87.890	151	158	106.127
10000	161	164	39.056	146	151	52.750	151	155	65.330	141	146	89.344	153	159	105.597
100000	160	165	37.456	148	153	48.196	150	154	62.155	143	148	81.626	152	158	103.523
fastPHASE	78	–	3002.853												

Table 4. Error counts for haplotyping and genotype imputation for a 10K dataset with 30 genotypes

Flanking markers	6			7			8			9			10		
λ	C_m	C_s	Time (s)	C_m	C_s	Time (s)	C_m	C_s	Time (s)	C_m	C_s	Time (s)	C_m	C_s	Time (s)
0	185	5656	98.151	178	5164	62.208	175	4878	159.304	170	4797	198.417	174	4534	225.164
10	186	5573	43.475	179	5137	100.959	167	4913	134.961	165	4695	175.310	174	4454	201.380
100	175	5294	31.619	164	4755	75.011	154	4528	89.023	158	4391	120.290	157	**3971**	144.147
1000	172	5223	22.828	150	4769	55.079	153	4443	71.537	**148**	4209	88.096	161	3997	116.173
10000	171	5277	21.457	151	4784	52.913	155	4494	67.211	149	4266	83.994	166	4086	110.170
100000	168	5295	20.940	152	4846	49.450	152	4506	65.101	155	4322	80.946	171	4146	104.924
fastPHASE	79	2070	3725.013												

Haplotyping also decreases computation times because it takes advantage of the phase information to the left of the central marker. For almost every value of λ, the MM algorithm outperforms the EM algorithm in phasing, genotype imputation and speed. Although fastPHASE makes fewer mistakes than Mendel, it is slower by one to two orders of magnitude, depending on parameter settings.

4 DISCUSSION

The EM algorithm has long served as a computational engine in haplotyping schemes. Our analysis demonstrates that penalization improves haplotype frequency estimation, genotype imputation, and haplotyping. In essence, penalization captures the parsimony nature imposes. The MM implementation of penalized estimation converges in fewer iterations than EM. Combining the MM algorithm with haplotype extension–elimination along a sliding window of markers makes it possible to handle hundreds of thousands of markers efficiently. Overall computational complexity scales linearly in the number of markers. The software described here will be made available to the public in the next release of Mendel.

Although our combination of methods does not lead to the lowest error rates in imputing missing genotypes, it is not clear that this is a serious handicap. If we accept 1% missing data as reasonable on the best genotyping platforms, then Mendel's overall error rate of 1/2000 should lead to very few incorrect inferences in association studies. The vast majority of errors committed still get one of the two alleles correct, and errors are less damaging in association studies than they are in linkage studies. This optimistic attitude should not be equated with complacency. Every source of error should be attacked.

The alternative to haplotyping via linkage disequilibrium is haplotyping via Mendelian inferences in pedigrees. When pedigree data are available, the two methods can be combined. The obvious tactic is to apply genotype elimination first marker by marker (Lange and Goradia, 1987). The partial phase information gleaned can then guide haplotype frequency estimation and genotype imputation, treating the genotyped pedigree members as unrelated. One anticipated problem with this approach is that the first stage may uncover genetic inconsistencies. In this rare circumstance, we suggest ignoring stage one and proceeding directly to haplotype frequency estimation. Neither haplotype frequency estimation nor Mendelian inferences depend on allele frequencies or map distances.

At the expense of more complex programming, there are several options for improving the speed and accuracy of the MM algorithm. For instance, one can chose window widths to reflect the local extent of linkage disequilibrium. Long windows are more compatible with strong linkage disequilibrium. We have used a fairly strict convergence criterion. Relaxing it cuts the number of iterations until convergence. There are obvious tradeoffs between speed and accuracy. Since there is no guarantee that the objective function is concave in penalized estimation, one can safeguard estimation by trying several different starting points. This tactic obviously increases computational times.

Although penalized estimation by itself does not lead to the most accurate haplotyping, it is important to stress its potential in more sophisticated schemes of haplotyping. It has much to offer in the more complex algorithms incorporated in fastPHASE. Bayesian estimation based on Markov chain Monte Carlo is less likely to be a beneficiary. Finally, it is worth mentioning that penalized estimation is apt to pay dividends in other areas of population genetics. High-dimensional estimation problems are here to stay in genetics, and one of our first reflexes in solving them should be to consider parameter regularization.

ACKNOWLEDGEMENTS

Funding: This research supported in part by National Institutes of Health (HG02536 to K.L.A.); United States Public Health Service (GM53275 to K.L., MH59490 to K.L.).

Conflict of Interest: none declared.

REFERENCES

Akey,J. *et al.* (2001) Haplotypes vs single marker linkage disequilibrium tests: what do we gain? *Eur. J. Hum. Genet.*, **9**, 291–300.

Ayers,K.L. *et al.* (2007) A dictionary model for haplotyping, genotype calling, and association testing. *Genet. Epi.*, **31**, 672–683.

Chen,S.S. *et al.* (1998) Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, **20**, 33–61.

Claerbout,J.F. and Muir,F. (1973) Robust modeling with erratic data. *Geophysics*, **38**, 826–844.

De Leeuw,J. and Heiser,W.J. (1977) *Geometric Representations of Relational Data*. Mathesis Press, Ann Arbor, MI.

Excoffier,L. and Slatkin,M. (1995) Maximum likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.*, **12**, 921–927.

Groenen,P.J. (1993) *The Majorization Approach to Multidimensional Scaling: Some Problems and Extensions*. DSWO Press, Leiden, The Netherlands.

Halperin,E. and Eskin,E. (2004) Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, **20**, 1842–1849.

Hawley,M.E. and Kidd,K.K. (1995) Haplo: a program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *J. Hered.*, **86**, 409–411.

Hunter,D.R. and Lange,K. (2004) A tutorial on MM algorithms. *Am. Stat.*, **58**, 30–37.

Lange,K. (2002) *Mathematical and Statistical Methods for Genetic Analysis*. Spring-Verlag, New York.

Lange,K. (2004) *Optimization*. Spring-Verlag, New York.

Lange,K. and Goradia,T.M. (1987) An algorithm for automatic genotype elimination. *Am. J. Hum. Genet.*, **40**, 250–256.

Lange,K. *et al.* (2001) Mendel version 4.0: a complete package for the exact genetic analysis of discrete traits in pedigree and population data sets. *Am. J. Hum. Genet.*, **69** (Suppl. 4), A1886.

Long,J.C. *et al.* (1995) An E-M algorithm and testing strategy for multiple-locus haplotypes. *Am. J. Hum. Genet.*, **56**, 225–232.

Marchini,J. *et al.* (2006) A comparison of phasing algorithms for trios and unrelated individuals. *Am. J. Hum. Genet.*, **78**, 437–450.

Qin,Z.S. *et al.* (2002) Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *Am. J. Hum. Genet.*, **71**, 1242–1247.

Santosa,F. and Symes,W.W. (1986) Linear inversion of band-limited reflection seismograms. *SIAM J. Sci. Stat. Comput.*, **7**, 1307–1330.

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.

Stephens,M. and Scheet,P. (2005) Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *Am. J. Hum. Genet.*, **76**, 449–462.

Stephens,M. *et al.* (2001) A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.*, **68**, 978–989.

Taylor,H.L. *et al.* (1979) Deconvolution with the ℓ_1 norm. *Geophysics*, **44**, 39–52.

Tibshirani,R. (1996) Regression shrinkage and selection via the Lasso. *JRSS-B*, **58**, 267–288.