**ABSTRACT**

We develop an hidden Markov model (HMM)-based algorithm for computing exact parametric and non-parametric linkage scores in larger pedigrees than was possible before. The algorithm is applicable whenever there are chains of persons in the pedigree with no genetic measurements and with unknown affection status. The algorithm is based on shrinking the state space of the HMM considerably using such chains. In a two g-degree cousins pedigree the reduction drops the state space from being exponential in g to being linear in g. For a Finnish family in which two affected children suffer from a rare cold-inducing sweating syndrome, we were able to reduce the state space by more than five orders of magnitude from $2^{27}$ to $2^{5}$. In another pedigree of state-space size of $2^{21}$, used for a study of pituitary adenoma, the state space reduced by a factor of 8.5 and consequently exact linkage scores can now be computed, rather than approximated.

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

1 INTRODUCTION

Genetic linkage analysis seeks to locate genomic regions that are likely to contain genes that increase the probability of traits such as heredity diseases. The input to such analysis are pedigrees of families that segregate a disease, marker information such as SNP readings, and affection status of some or all family members. The main idea is that markers which are found in the same vicinity on the chromosome are more likely to stay together during meiosis. Thus, based on the topology of the pedigree and the marker readings, it is possible to compute how likely it is for a predisposing gene to be together selector variables that control the inheritance from a founder allele in the pedigree, indicating only if the allele is transmitted to the individual at the end of the chain or, if not, what is the number of selectors that block its inheritance. For $r$ selectors (meiosis), this clustering reduces the state space from $2^{g}$ to $r+1$ states.

To demonstrate the usefulness of the state-space reduction, we provide several examples of pedigrees in which computations are significantly easier once the reduction is used. For a Finnish family in which two affected children suffer from a rare cold-inducing sweating syndrome (Knappskog et al., 2003), we were able to reduce the state space of the internal inbreeding loop by over 9-folds, and to reduce the state space for the entire pedigree by more than five orders of magnitude from $2^{20}$ to $2^{3}$. For another pedigree, recently used for the study of pituitary adenoma (Vierimaa et al., 2006), we were able to reduce the state space from $2^{21}$ to $60 \cdot 2^{13}$, by a factor of more than 8.5. Previously, only approximate scores could be computed for this pedigree on a standard PC and software for exact linkage computations such as Merlin (Abecasis et al., 2002) failed as reported by Albers et al. (2008). Our space reduction enables exact linkage computations and we demonstrate order of magnitude experimental speedups when performing computations across 6000 genomic locatons, as used by standard SNP panels for linkage analysis.

© 2009 The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

To whom correspondence should be addressed.

**Supplementary data are available at Bioinformatics online.**
2 STATE REDUCTION IN HMMS

Consider an HMM with hidden variables $S_i$ and observed variables $X_i$, $i = 1, \ldots, L$ (Rabiner and Juang, 1986). The (hidden) state space is the set $S$ of possible values for $S_i$. The state space is identical for every slot $i$. The likelihood of data $(x_1, \ldots, x_L)$ for $L$ slots is specified via two main components. The single slot likelihood of data $P(x_i | s_i)$ at slot $i$ given a state $s_i$ for $S_i$ and the transition probabilities $P(S_i | s_i)$ for a class state $s_i$ from a state $s_{i-1}$ at slot $i-1$ to slot $i$.

$$P(data) = \sum_{s_i} P(x_i | s_i) P(s_i | s_{i-1})$$

$$P(S_2 | s_1) = \sum_{s_2} P(x_2 | s_2) P(s_2 | s_1)$$

$$P(S_L | s_{L-1}) = \sum_{s_L} P(x_L | s_L) P(s_L | s_{L-1})$$

The time complexity of computing this sum grows quadratically with the size of the state space $|S|$ and linearly in the number of slots $L$. The time complexity is $O(|S|^2 L + c |S|)$ where $c$ is an upper bound for computing the single slot likelihood $P(s_i | s_{i-1})$. We note that in many HMM applications, including linkage analysis, the goal is to compute the marginal probabilities $P(S_i | x_1, \ldots, x_L)$ for all $i = 1, \ldots, L$ rather than to compute the likelihood of data. This task can be completed using the junction-tree algorithm with only twice the computational cost (Lauritzen and Spiegelhalter, 1988). We show experimental results for both tasks, but restrict discussion of the complexity of computing to the likelihood itself to simplify presentation.

We focus on applications where $S$ is possibly very large such as for linkage analysis where it grows exponentially in, roughly, the number of persons in the pedigree. In such cases, the dominating factor $|S|^2$ can be reduced substantially if the state space $S$ can be partitioned into equivalence classes $[s]$ for which the likelihood of data is constant. This effectively changes the sum over the state space at each slot to a sum over equivalence classes. The dominating complexity will now depend on the number of equivalence classes rather than on the number of states in $S$.

The likelihood is computed for one representative of each equivalence class via

$$P(data) = \sum_{[s]} P(x | [s]) P([s] | [s_{i-1}])$$

$$P(S_2 | [s_1]) = \sum_{[s_2]} P(x_2 | [s_2]) P([s_2] | [s_1])$$

$$P(S_L | [s_{L-1}]) = \sum_{[s_L]} P(x_L | [s_L]) P([s_L] | [s_{L-1}])$$

(2)

where the prior for a class $[s]$ is the sum over the priors of its constituent states, namely, $P([s]) = \sum_{s \in [s]} P(s)$.

The equivalence of Equations (1) and (2) stems from two general conditions. Explicating these conditions facilitates the discovery of novel equivalence classes that reduce the computational cost of likelihood computations, as we show for genetic linkage analysis.

Condition I: the single-slot likelihood given a hidden state $s$ is equal for all states in the equivalence class $[s]$, namely, $P(x_i | s_i) = P(x_i | [s])$ for all $s$ and $s'$ in the same equivalence class.

Hence, we can safely define the single-slot likelihood given an equivalence class via $P(x_i | [s]) = P(x_i | s_i)$.

Condition II: denote by $P([s] | s') = \sum_{s \in [s]} P(s | s')$ the transition probability from state $s'$ to an equivalence class $[s]$. The condition is that this transition probability does not distinguish between states in the same equivalence class, namely, $P([s] | s') = P([s'] | s')$ for all $s'$ and $s''$ in the same equivalence class. Hence, we can safely define the transition probabilities between equivalence classes via $P([s'] | [s]) = P([s'] | s)$.

These two natural conditions are sufficient to ensure that Equations (1) and (2) are equivalent due to the following reasoning. We first rewrite the right most sum in Equation (1). The following equality is due to Condition I.

$$\sum_{s_i} P(S_i = s_i | S_{i-1} = s_{i-1}) P(x_i | S_i = s_i)$$

$$= \sum_{s_i} P(S_i = s_i | S_{i-1} = s_{i-1}) P(x_i | S_i = [s_i])$$

The latter sum is further rewritten.

$$\sum_{[s_i]} \left[ \sum_{s \in [s_i]} P(S_i = s_i | S_{i-1} = s_{i-1}) P(x_i | S_i = [s_i]) \right]$$

$$= \sum_{[s_i]} P(S_i = s_i | S_{i-1} = s_{i-1}) P(x_i | S_i = [s_i])$$

where the final equality is due to Condition II. Proceeding with these steps over decreasing indices $L, L-1, \ldots, 1$ transforms Equation (1) to Equation (2) where in the last step $P([s_i])$ is set to $\sum_{s \in [s_i]} P(s)$.

3 STATE-SPACE REDUCTION IN FACTORED HMMS

Factored HMMS (Ghahramani and Jordan, 1997) are HMMs in which the hidden variables are a vector $S = (S_1, \ldots, S_k)$ with values drawn from a Cartesian product $H_1 \times \cdots \times H_k$ and with a transition probability defined component by component for each state $s_i$ in $S_i = (s_1, \ldots, s_k)$ as

$$P(S_i = (s_1, \ldots, s_k) | S_{i-1} = (s'_1, \ldots, s'_{i-1})) = \prod_{j=1}^{k} P_j(s'_j | s_j)$$

and for the first slot, $P(S_1 = (s_1, \ldots, s_k)) = \prod_{j=1}^{k} P_j(s'_1)$. When all the component transition probabilities $P_j$ are equal for all $j$, we term the resulting HMM a homogeneously factored HMM.

Factored HMMS offer computational benefits when computing the likelihood of data. Ghahramani and Jordan (1997) show how specifying the probabilities $P(S_i | S_{i-1})$ via a product as in Equation (3) reduces the time complexity to $O(|S||\log |S| + c |L||S|)$. Their algorithm is a special case of bucket elimination (Dechter, 1998). We note that computing the $L$ marginal probabilities $P(S_i | x_1, \ldots, x_L)$ in a factored HMM can be performed with only twice the amount of computations using the junction-tree algorithm (Lauritzen and Spiegelhalter, 1988).

We offer state-space reductions for homogeneously factored HMMS that maintain these benefits and further reduce the computational complexity. For simplicity of notation, we assume that $H_i = [0, 1]$ and so each $s_j$ is a binary variable, which we call a selector, and the state-space size is $2^k$. The state-space reductions
are formed by clustering the selectors and partitioning the states of each cluster so that Conditions I and II are satisfied.

Assuming the choice of clusters is such that Conditions I and II are satisfied, we now explicate how the likelihood of data computations are carried out in the reduced state space, and examine the reduction in time and space complexity of the computation. In particular, starting with Equation (1), we need to show how the computation of each sum is done when the state space is in a factored form. Suppose \( B = \{ B_1, \ldots, B_n \} \) is a set of disjoint clusters of all selectors for some slot \( t \) with \( r_1, \ldots, r_m \) selectors, respectively, and suppose \( A = \{ A_1, \ldots, A_{\bar{M}} \} \) is a set of disjoint clusters of all selectors for the previous slot where \( B_i \in B \) iff \( \sum_{j=1}^{l-1} E_{A_j} \). Let \( A_2 \) and \( B_i \) denote vectors of zeros and ones of length \( r_t \). Then the final sum in Equation (1), denoted by \( \Sigma_2 \), can be written as follows.

\[
\Sigma_2 = \sum_{b_1, \ldots, b_n} P(\{ b_t \} | a_t) - P(b_m | a_n). \]

Due to Condition I, we get,

\[
\Sigma_2 = \sum_{b_1, \ldots, b_{n-1}} P(\{ b_t \} | a_t) - P(b_m | a_{n-1}).
\]

Due to Condition II, the last sum equals \( P(b_m | a_n) \). Incorporating these two modifications sequentially for the indices \( m, m-1, \ldots, 1 \) yields

\[
\Sigma_2 = \sum_{b_1} P(\{ b_1 \} | a_t) - \sum_{b_3} \sum_{b_2} P(\{ b_2 \} | a_n). \tag{4}
\]

This sum is carried out right to left, summing over \( b_m \), then over \( b_{m-1} \) and finally over \( b_1 \). The result is a conditional probability table \( \Sigma_2 = \lambda(\tau_t | a_t, \ldots, a_m) \). This conditional probability table is carried to the \( L - 1 \) sum of Equation (2), and this process is repeated \( L \) times, once per slot. Hence, the likelihood depends on the cluster states and not on the states of individual selectors.

We now define a specific choice of clusters and prove that it satisfies Condition II. In the next section, we specify additional domain-specific restrictions in order for this choice to also satisfy Condition I, as needed in order to achieve computational savings. For simplicity, we assume a symmetric transition probability table, which is the case that a transition from state 0 at slot \( t \) to state 1 and from state 1 to 0 are equal and are denoted by \( \theta_t \). Note that the results can be easily extended beyond binary domains \( H_t = \{0,1\} \) and without assuming symmetric transition probability tables, but this extension is not needed for genetic linkage analysis. A selector can have two complement states: ON and OFF. For a cluster \( C \) with \( r \) selectors, a state \([i]\) of the reduced state space of \( C \) is the equivalence class which contains all vectors of size \( r \) that have \( j \) entries being \( 0 \) and \( r-j \) being \( 1 \). For example, \( 010, 001, 100 \) are equivalent under the \( \theta \) operation. Thus, we have \( c(j,r)=r!/j!(r-j)! \) vectors in state \([i]\) for \( j=0,1,\ldots,r \). This set of \( r+1 \) equivalent classes is called the counting partition.

\textbf{THEOREM 1.} Let \( S = (S^1, \ldots, S^m) \) be a vector of selectors and let \( C = \{ C_1, \ldots, C_m \} \) be a set of disjoint clusters with \( r_1, \ldots, r_m \) selectors, respectively, in each cluster, where \( k = \sum_{j=1}^{m} r_j \). Then a factored HMM in which the hidden variable has values drawn from the Cartesian product \( \Gamma = \prod_{j=1}^{m} \Gamma_j \times \cdots \times \Gamma_m \), where \( \Gamma_j \) is the set of equivalence classes of cluster \( C_j \) generated by using the counting partition, satisfies Condition II.

\textbf{PROOF.} To prove that Condition II holds for the counting partition, we consider a single cluster \( C \in \Gamma \) with \( r \) selectors. The transition probability \( P(i | j) \) for switching from a state \( \bar{c}_i \) of \( C \) with \( j \) positions \( \theta \neq \theta \) on to any one state with \( i \) positions \( \theta \) is developed below. Let \( \theta \) be the probability of switching from state \( \theta \) to state \( \theta \) and of switching from state \( \theta \) to state \( \theta \). The other two transitions have probability \( 1 - \theta \). The probability of switching from a state \( \bar{c}_i \) where \( j \) selectors are \( \theta \) on to the state \( [j] \) in which some arbitrary \( i \) selectors are \( \theta \) on is given by

\[
P(i | j) = \sum_{t=\max(0, i-j)}^{\min(i,j)} \theta_t (i-t) \theta_j (r-j-2t) \theta^{j-i-2t} \theta^{j-i-2t}. \tag{5}
\]

where \( r \) is the number of selectors that are \( \theta \) on both in \([i]\) and in the state \( \bar{c}_i \). Since this formula does not depend on which \( j \) selectors are \( \theta \), it follows that \( P(i | j) = P(i | j) \). This is exactly Condition II for one cluster of \( r \) selectors. These definitions of the transition probability tables apply separately to each of the clusters \( C_1, \ldots, C_m \). Consequently, the conditional probabilities \( P(x | z) \) satisfy Condition II via

\[
P(x | z) = \prod_{l=1}^{m} P([c_l] | c'_l) = \prod_{l=1}^{m} P([c_l] | c''_l) = P(x | z'),
\]

where \( c_j \) is the component of state \( j \) for the selectors associated with \( C_j \), and where \( c'_j \) and \( c''_j \) are the components for the selectors \( C_j \) of the two equivalent states \( ' \) and \( " \).}

The counting partition reduces the state space of each cluster with \( r \) selectors from \( 2^r \) to \( r+1 \) states. Thus, the complexity of computing Equation (4) for this partition is the following. Suppose the \( k \) selectors are divided into \( m \) equally sized clusters each corresponding to \( k/m \) selectors and having \( d' = k/m \) states. Then summing over \( b_m \) yields a probability table \( \lambda(\tau_t | a_t, \ldots, a_{m-1}) \). The next sum yields a table \( \lambda(\tau_t | a_t, \ldots, a_{m-2}) \). Finally, the conditional probability table \( \Sigma_2 = \lambda(\tau_t | a_t, \ldots, a_1) \) is created. Since each \( \lambda \) table has \( m \) dimensions, each step involves \( O(d'm^{m-1}) \) arithmetic operations. This step is repeated \( m \) times, and therefore \( O(d'm^{m-1}) \) arithmetic operations are used. The entire process is repeated \( L \) times, once per slot, and so the overall complexity is \( O(Lm^{(1+k/m)^m}) \) where \( 1 \leq m \leq k \). For example, if \( m = k \), each cluster contains one selector, the complexity is \( O(L2^{k+1}) \) as suggested by Ghahramani and Jordan (1997) and used in all the current HMM-based linkage analysis programs (Kruglyak and Lander, 1996; Markianos et al., 2001). On the other extreme, when all selectors are clustered together, namely \( m = 1 \), then the complexity is merely \( O(L(1+k)) \).

Examples of utilizing the counting partition and obtaining significant speed up in real genetic applications are discussed in the next two sections.
4 APPLICATION IN GENETIC LINKAGE ANALYSIS

The purpose of genetic linkage analysis is to score the human genome in such a way that produces high scores for areas that harbor genes that predispose to a disease under study. The means are pedigrees of families that segregate the disease and genetic information such as SNP data measured on individual members of these families. There are various scoring methods for linkage analysis, some are called parametric and some non-parametric, but all scoring methods share the backbone of a common HMM. This HMM is in fact a homogeneously factored HMM and its state space is defined by a set of selectors precisely as discussed in the previous section. The data at slot $i$ are the measurements of the individuals’ genetic material at the $i$-th location.

In this section, we provide the needed background to describe the meaning of the transition probabilities, define precisely the likelihood $P(x_i | y_i)$ of data at slot $i$ given a hidden state and provide clustering methods that generate the space-state reductions via the counting partition as studied in the previous section.

4.1 HMM for linkage analysis

A pedigree is a directed acyclic graph $(V, E)$ with a set $V$ of vertices of two possible types called male and female and a set of directed edges $E \subseteq V \times V$ such that for each vertex $v$ there is at most one directed edge $(u, v)$ for which the type of $u$ is male and at most one edge for which the type of $v$ is female. When a vertex has in-degree 0, it is called a founder. A vertex that is not a founder is called a non-founder. Each vertex in a pedigree is classified either as typed (measured) or untyped (not measured).

Semantically, each vertex in a pedigree represents a person and each directed link represents a parent–child relationship. When a vertex has in-degree 1, it means that one parent is specified in the pedigree and the other is not. A typed vertex represents a person whose genetic material has been measured. Such person is also said to be measured.

Definition. A potential descent graph for a pedigree $(V, E)$ is a directed acyclic graph $(V', E')$ such that for every vertex $v' \in V'$, there are two vertices $m_{ij}$ (termed the maternal vertex) and $p_{ij}$ (termed the paternal vertex) in $V'$, and for every edge $(v_i, v_j) \in E$ there are two edges in $E'$ as follows: if the type of vertex $v_i$ is male, then the edges $(m_{ij}, v_j)$ and $(p_{ij}, v_j)$ are in $E'$ and if the type of vertex $v_j$ is female, then the edges $(m_{ij}, v_i)$ and $(v_i, p_{ij})$ are in $E'$. If a vertex $v_i \in V$ is typed, then both $m_{ij}$ and $p_{ij}$ are typed and if $v_j$ is not typed then both $m_{ij}$ and $p_{ij}$ are untyped.

Semantically, the meaning of a pair of vertices $(m_{ij}, p_{ij})$ is the maternally inherited and paternally inherited genetic information of person $j$ at some genomic location. A parent $i$ contributes either $m_{ij}$ or $p_{ij}$ to each child $j$.

Definition (Sobel and Lange, 1996). A descent graph $D=(V', E')$ of a potential descent graph $(V', E')$ is a subgraph of $(V', E')$ such that $V''=V'$ and for every pair of edges $[(m_{ij}, p_{ij})] \in E'$ exactly one is in $E'$ and for every pair of edges $[(m_{ij}, m_{ij})]$ in $E'$ exactly one is in $E''$.

Vertices that are classified as typed in $V''$ remain typed in $V'$ and the other remain untyped.

Note that for a pedigree with $n$ non-founders there are $2^n$ descent graphs, since there is a binary choice of genetic material twice for every person that is not a founder. For each choice from a pair of edges, we assign a binary variable called a selector whose values are 0 if the first edge from a pair is chosen and 1 otherwise. The vector of selectors, which is called the inheritance vector, can get $2^n$ assignments and each assignment $s$ defines a descent graph denoted by $D[s]$. Each assignment $s$ is called an inheritance state.

Each descent graph specifies how each of the founding alleles is inherited. That is, given a descent graph of a pedigree and an assignments of alleles $a=(a_1,\ldots,a_f)$ to the maternal and paternal variables of its $f$ founders, every maternal and paternal variable is assigned a specific founding allele. In other words, each descent graph consists of $2^f$ directed trees, two for each founder, called descent trees and each descent tree specifies how one founder allele is assigned to the maternal and paternal vertices that constitute that tree.

A label of a typed person $v_j$ is an unordered pair of letters $[a_i, a_j]$ from a finite set $A$. An element of $A$ is called an allele.

The label is also termed the genotype of person $v_j$ (at some genomic location). The marker data at some genomic location are a vector of genotypes—one for each typed person.

In the case of SNP marker data, there are only two letters in $A$ and hence each label has three options: $[0, 0], [1, 1]$, or $[0, 1]$. In the case of simple tandem repeats (STR) markers, there are $r$ letters in $A$ and therefore $2^r$ possible labels of the form $[a_1, a_2]$. If the genotype of a person is $[a', a'']$ then either its maternal allele is $a'$ and its paternal allele is $a''$ or the converse. The marker data are a vector of unordered pairs and does not distinguish which allele is the maternal and which is the paternal. Each typed person $v_j$ adds a constraint on the possible alleles for $(m_{ij}, p_{ij})$.

Definition. A vector of founder alleles $a$ is said to be consistent with marker data $x_i$ and a descent graph $D[s]$ iff $x_i$ can be obtained by inheritance via $D[s]$ from $a$. This consistency statement is denoted by $a \Rightarrow x_i \wedge s$.

The HMM for linkage analysis can now be defined as in Sobel and Lange (1996). The likelihood of a marker data vector $x_i$ given a state $s$ of the inheritance vector is specified by

$$P(x_i | s) = \sum_{a} P(a) \cdot \prod_{i=1}^{2^n} P(a_j)$$

where $P(a)$ equals the probability of the founders having a vector of founder alleles $a=(a_1,\ldots,a_f)$. The product form is justified by the common assumption that the founders are random persons from a population and their two alleles are randomly sampled as well (called Hardy–Weinberg equilibrium). As written, this sum is exponential in the number of founders. However, Sobel and Lange (1996) devised an efficient polynomial algorithm for this sum using founder graphs.

The transition probabilities are given by

$$P(S_i=(s_1,\ldots,s_f) | S_{i-1}=(s_1',\ldots,s_{f-1}')) = \prod_{j=1}^{f} P_j(s_j' | s_{j-1}'),$$

where $P_j(s_j' | s_{j-1}')=0$ if $s_j' \neq s_{j-1}'$. The biological meaning of the statement $s_j' \neq s_{j-1}'$ is that a recombination has occurred in the $j$-th meiosis, namely, in one genomic location a maternal allele of a parent is transmitted to a child and in the next location the paternal allele of the same parent is transmitted to the same child.

The specified model is a homogeneously factored HMM and therefore any clustering of the selectors using the counting
partitioning satisfies Condition II, as shown in Theorem 1. The remaining challenge is to define clusters that also satisfy Condition I and for this the properties of the likelihood \( P(x_i) \) [Equation (5)] must be studied in detail.

4.2 Chain reductions

The main idea for identifying useful clusters is to find chains in the pedigree such that either the chain is on, meaning that an allele is transmitted from the start of the chain to its end, or the chain is off, in which case a random allele is transmitted to the end of the chain. Clusters of such chains only depend on the number of selectors that block the transmission of that allele. This idea is made precise as follows.

**Definition.** A chain of length \( l \) in a pedigree is a sequence of edges \((v_1, v_{i+1}), i = 1, \ldots, l\), such that nodes \( v_1, \ldots, v_l \) are each untyped and have one incoming edge and one outgoing edge in the pedigree, and node \( v_{i+1} \) has one incoming edge (but may or may not be typed and may have any number of children).

A chain of length \( l \) in a pedigree translates to a set of \( 2l \) edges in the potential descent graph of the pedigree connecting the maternal and paternal nodes of person \( v_l \) to the paternal node of person \( v_{i+1} \) when \( v_l \) is a male, and connecting them to the maternal node of \( v_{i+1} \) when \( v_l \) is a female. In the potential descent graph, we define the maternal node \( m_l \) to be the source of the chain if the parent of person \( v_l \) in the pedigree is a female and define the paternal node \( p_l \) as the source, if the parent of person \( v_l \) is a male. Similarly, we define the sink of the chain to be node \( m_{i+1} \) if person \( v_l \) is a female, and define node \( p_{i+1} \) as the sink, if person \( v_l \) is a male.

There is one descent graph in which the source of the chain is connected by a directed path to the sink of the chain. Let \( S_1, \ldots, S^l \) be the selectors associated with the \( l \) choices of these edges and define an on state as the choice of an edge on the directed path from source to sink, and by off a choice that disconnects this path.

In practice, before searching for chains in a given pedigree, we transform the pedigree to a normalized form by using the following two operations repeatedly until they no longer apply. First, remove an untyped parent that has no children. Second, remove an untyped founder that has one child. It can be shown that the likelihood of marker data remains unchanged under these transformations. Furthermore, it can be assumed that the pedigree is specified in a normalized form. This normalization procedure is merely a rule that tells the geneticist when there is no need to add more persons to the pedigree specification, which, in principle, can expand endlessly to various directions. The normalization procedure can create more chains and consequently facilitates larger state-space reductions.

For example, consider Pedigree I depicted in Figure 1 with three typed children \( D_t, E_u \) and \( F_t \) who are distant cousins. The parents of individuals along the chains \( D, E \) and \( F \) are not specified in the normalized pedigree. For each chain \( C_j \), we define a cluster which we denote also by \( C_j \).

The cluster \( C_j \) consists of the selectors associated with the chain \( C_j \). Such a chain cluster with \( l \) selectors can get one of \( l+1 \) values, corresponding to the number of selectors that are on. This reduction is the counting partition, defined in Section 3. The following theorem states that such chain reductions do not change the likelihood of data. The proof is given in the appendix of the Supplementary Material.

**Theorem 2.** Let \( S_1, \ldots, S^l \) be the selectors for a pedigree \((V, E)\) and let \( C \) be a chain in \((V, E)\) such that \( S^{l_1}, \ldots, S^{l_2} \) are the selectors associated with the edges of \( C \). Let \( S^{l_1} \) be a variable with a value equal to the number of selectors \( S^l \) that are on for \( 1 \leq j \leq 1. \) Then the likelihood of data can be computed by summing over the states of \( S^{l_1}, S^{l_2+1}, \ldots, S^l \).

This chain reduction can be repeated for every chain in the pedigree yielding considerable reduction in the state space. For example, the full-inheritance state space of Pedigree I in Figure 1 corresponds to \( 4^{t+u+v+z} \) informative meiosis yielding a state space of \( N = 2^{t+u+v+z+4} \). The reduced state space uses the fact that the likelihood of data does not depend on the exact state of the selectors along the chains \( D, E \) and \( F \) that connect the typed cousins to the two founders, but only on whether they point toward a common ancestor, and if not, how many selectors do not point to the common ancestor. In other words, how many selectors are off. Consequently, for each \( W \)-chain, where \( W \in \{D, E, F\} \) of length \( w \), it is possible to cluster the \( w-1 \) selectors from \( W \) along the chain into a single variable with \( w \) states. Thus, the total reduced state space is now \( N' = 2^{w} \), yielding an exponential reduction of the state space.

5 EXPERIMENTAL RESULTS

We demonstrate the power of the state space reduction in factored HMMs for genetic linkage analysis problems via the pedigree depicted in Figure 2. This pedigree was recently used for the study of...
Speeding up HMM algorithms for genetic linkage analysis

Fig. 2. A normalized pedigree used by Vierimaa et al. (2006) for the study of pituitary adenoma. There are six typed individuals in the pedigree that are marked in black stripes. We use chains 1, 2 and 3 to reduce the state space by a factor of 8.5 and speedup computations.

Fig. 3. Runtime comparison for computations using the original model and the reduced state-space model for the pedigree in Figure 2 as a function of the length \( m \) of Chain 1.

Fig. 4. Runtime comparison for computations using the original model and the reduced state-space model for the pedigree in Figure 2 across 6000 markers, where Chain 1 is of length 0.

Pituitary adenoma (Vierimaa et al., 2006). Since the state-space size of the pedigree is \( 2^{27} \), exact linkage scores could not be computed and heuristics were used (Albers et al., 2008). Reducing the chains marked in the figure, of lengths 2, 3 and 4, reduces the state space to \( 2^{18} \times 3 \times 4 \times 5 \) by an overall factor of 8.5. Consequently, exact linkage scores can now be computed, rather than approximated.

We implemented an algorithm that computes the probabilities \( P(S_i|\text{data}) \) for every location \( i \), which facilitates the computations of parametric and non-parametric linkage scores. Our implementation supports the factored HMM model for genetic linkage analysis with and without the state-space reduction. We used this software to compare the runtime of computations in the reduced model with the runtime of the original model across 6000 markers, for variable lengths of the chain marked as ‘Chain-1’. Figure 3 shows the runtime in hours for the two models for lengths \( m=0, \ldots, 4 \). For \( m=3 \) and \( m=4 \) the time it takes to perform computations on the original model has been extrapolated from running the software for this model across 150 locations and multiplying by 40. It is evident from the figure that the runtime grows linearly in \( m \) in the new model, while it is exponential in \( m \) in the original model. The probabilities computed are the same in both models. In addition, Figure 4 plots the runtime of the two models for \( m=0 \) as a function of the number of markers for the computation of the likelihood of data and the inheritance probabilities at all loci given the data. As can be seen, the runtime is linear in the number of markers for both models and the runtime ratio is maintained regardless of the length of the model, as expected. In addition, as predicted from complexity analysis, the runtime of computing probabilities at all loci is twice that of computing the likelihood of data.

6 DISCUSSION

In this article, we described two general conditions which, when satisfied, allows one to reduce the state space of HMMs and factored HMMs. We also described when these conditions can be applied to linkage analysis problems which yields a new method for performing exact linkage computations at a potentially reduced cost. In general, when our method reduces the size of the state space it yields a computational savings and, for linkage problems, these savings are exponential.

The computation of exact linkage scores (LOD or non-parametric) in linkage analysis is of great importance. Having an exact linkage scores provides researchers confidence to proceed with the often expensive and time-consuming fine-mapping process. The use of approximate linkage score methods typically yields no guarantees or loose bounds that do not enable a researcher to draw conclusions regarding the linkage score. In addition, for stochastic-based Monte Carlo techniques, researcher must rely on approximate tests of convergence.

Our state-space reduction method does not yield a computational benefit for all pedigrees, but, as we describe herein, it can yield a significant reduction of the computational cost. This reduction can be crucial for some pedigrees, turning a previously intractable computation into one that is tractable, and beneficial for others.
than having one cluster per chain. The state space reduces from counting variable with 14 states that replaces both chains, rather than children and one selector for a child of the common founder. All affected individuals. We retain four selectors for the two affected family shown in Figure 5 which connects the two parents of the state space, the likelihood of data depends only on the combined N states. The total reduced state space dropped from 250, which is completely infeasible, to a state space of $2^{52}$, a reduction of more than five orders of magnitude.

HMMs and factored HMMs are widely used in various applications, thus speeding up common algorithms in these models, as done via the state-space reduction, can be proved useful for other domains as well. We note that the challenge in applying the reduction to other domains lies in finding a suitable partition of the state space, which satisfies Conditions I and II. Once these conditions are satisfied, the computational savings are automatic and do not require a special analysis. Finally, we note that our state-space reductions are also immediately applicable to other methods for linkage analysis such as Silberstein et al. (2006), Sobel and Lange (1996) and Thompson (1994).

Acknowledgements

We thank Edward Vitkin for an initial implementation of the reported algorithms.

Conflict of Interest: none declared.

References

Abecasis,G.R. et al. (2002) Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. Nat. Genet., 30, 97–101.

Albers,C. et al. (2008) Multipoint approximations of identity-by-descent probabilities for accurate linkage analysis of distantly related individuals. Am. J. Hum. Genet., 82, 607–622.

Cuttingham,R.W. et al. (1993) Faster sequential genetic linkage computations. Am. J. Hum. Genet., 53, 252–263.

Dechtere,R. (1998) Bucket elimination: a unifying framework for probabilistic inference. In Jordan,M.I. (ed.) Learning in Graphical Models. Kluwer Academic Press, pp. 75–104.

Elston,R. and Stewart,J. (1971) A general model for the analysis of pedigree data. Hum. Hered., 21, 523–542.

Fishelson,M. and Geiger,D. (2002) Exact genetic linkage computations for general pedigrees. Bioinformatics, 18(Suppl. 1), S189–S198.

Ghitzaumen,Z. and Jordan,M. (1997) Factorial hidden Markov models. In Machine Learning. MIT Press.

Gudbjartsson,D.F. et al. (2000) Allegro, a new computer program for multipoint linkage analysis. Nat. Genet., 25, 12–13.

Gudbjartsson,D. et al. (2005) Allegro version 2. Stat. Sci., 37, 1015–1016.

Ingelsoedtstra A. and Radda,D.G. (2005) Genetic linkage analysis, algorithms and their implementation. Trans. Comput. Syst. Biol., 123–144.

Knappskog,P. et al. (2003) Cold-induced sweating syndrome is caused by mutations in the CRLF1 Gene. Am. J. Hum. Genet., 72, 355–363.

Lander,E. and Green,D. (1998) Faster multipoint linkage analysis using Fourier transform. J. Comput. Biol., 5, 1–7.

Lander,E. and Kronyk,L. et al. (1995) Rapid multipoint linkage analysis of recessive traits in nuclear families including homozygosity mapping. Am. J. Hum. Genet., 56, 519–527.

Lander,E. et al. (1996) Parametric and nonparametric linkage analysis: a unified multipoint approach. Am J. Hum. Genet., 58, 1347–1363.

Lander,E. and Green,E. (1987) Construction of multilocus genetic maps in humans. Proc. Natl Acad. Sci., 84, 2363–2367.

Lange,K. (1997) Mathematical and Statistical Methods for Genetic Analysis. Springer, New York.

Lauritzen,S.L. and Spiegelhalter,D.J. (1988) Local computations with probabilities on graphical structures and their application to expert systems (with discussion). J. R. Stat. Soc. Ser. B, 50, 157–224.

Lauarkin,S.L. (1996) Graphical Models. Oxford University Press, Oxford.

Markianos,K. et al. (2001) Efficient multipoint linkage analysis through reduction of inheritance space. Am. J. Hum. Genet., 68, 963–977.

O'Connell,J. and Weeks,D. (1995) The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype soring and fuzzy inheritance. Nat. Genet., 11, 402–408.

Sobel,E. (1999) Analysis of Human Genetic Linkage. Johns Hopkins University Press, Baltimore, MD.
Speeding up HMM algorithms for genetic linkage analysis

Pearl.J. (1988) Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco, CA.

Rabiner,L.R. and Juang,B.H. (1986) An introduction to Hidden Markov models. IEEE ASSP Mag., 415.

Silberstein,M. et al. (2006) Online system for faster multipoint linkage analysis via parallel execution on thousands of personal computers. Am. J. Hum. Genet., 78, 922–935.

Sobel,E. and Lange, K. (1996) Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker sharing statistics. Am. J. Hum. Genet., 58, 1323–1337.

Thompson,E.A. (1994) Monte Carlo likelihood in genetic mapping. Stat. Sci., 9, 355–366.

Vierimaa,O. et al. (2006) Pituitary Adenoma predisposition caused by germline mutations in the AIP Gene. Science, 312, 1228–1230.