

Article

Learning Genetic Population Structures Using Minimization of Stochastic Complexity

Jukka Corander^{1,3}, Mats Gyllenberg¹ and Timo Koski^{2,*}

¹ Department of Mathematics and statistics, University of Helsinki, P.O.Box 68, FIN-00014 University of Helsinki, Finland

² Department of Mathematics, Royal Institute of Technology, S-100 44 Stockholm, Sweden

³ Department of Mathematics, Åbo Akademi University, FIN-20500 Åbo, Finland

* Author to whom correspondence should be addressed; E-Mail: tjtkoski@kth.se, Tel. +4687907134, Fax. +4687231788

Received: 21 February 2010 / Accepted: 28 April 2010 / Published: 5 May 2010

Abstract: Considerable research efforts have been devoted to probabilistic modeling of genetic population structures within the past decade. In particular, a wide spectrum of Bayesian models have been proposed for unlinked molecular marker data from diploid organisms. Here we derive a theoretical framework for learning genetic population structure of a haploid organism from bi-allelic markers for which potential patterns of dependence are a priori unknown and to be explicitly incorporated in the model. Our framework is based on the principle of minimizing stochastic complexity of an unsupervised classification under tree augmented factorization of the predictive data distribution. We discuss a fast implementation of the learning framework using deterministic algorithms.

Keywords: factorization of multivariate distributions; finite mixture models; Minimum Description Length; population genetics; statistical learning; structured population

1. Introduction

The concept of a *structured* or *subdivided* population has received intensive attention both in applied and theoretical population genetics for several decades. Intuitively, such populations can be considered to harbor multiple pools of individuals, each associated with distinct allele frequencies over a set of

molecular marker loci. For a mathematical conceptualization of such a structure in population genetics, see [1,2]. In a traditional formulation, statistical inference regarding the extent of subdivision that a population harbors is performed through summary statistics calculated from genotype data available for a number of local, geographically determined sample populations. Many different forms of such summary statistics exist in the literature; for details see the previously cited references. Despite the fact that the traditional formulation is still widely utilized in applied population genetics, a complementary approach to determining how subdivided a population is, has grown particularly popular within the most recent decade. This approach is fundamentally based on statistical learning of a mixture model for genotype data from multiple marker loci, where the mixture components are representing gene pools that have drifted apart over time. In a typical setting both the appropriate number and the contents of the mixture components are unknown *a priori* for a sample of individuals that have been genotyped. Currently, a large body of literature exists about the mixture-based inference of population structure, where numerous ramifications of a simple mixture model can be found [3–9]. A survey of the existing methods from a spatially explicit perspective can be found in [10].

We have earlier demonstrated how a Bayesian representation of statistical uncertainty related to population structure can be derived from a combination of generalized exchangeability and random urn models [7]. The operational interpretation of this model is that the observed genotypes and the alleles within each genotype are conditionally independent over all the considered loci, given the underlying allele frequency parameters in each latent pool of the total population. The latent pools themselves are represented by a stochastic partition of the sampled dataset, where the number of pools necessary for sufficiently expressive representation of the molecular variation is explicitly determined. In machine learning terminology the stochastic partitions perform unsupervised classification of data from multiple finite alphabets, such that the occurrence of the letters is considered to be conditionally independent over the alphabets given the classification of the items for which the letter combinations are observed.

In the current work we derive a generalization of unsupervised classification approach to inferring genetic population structures from biallelic multilocus data, where the loci are no longer assumed to be conditionally independent given a pool. The statistical learning approach is derived using information theoretic notions where the minimum description length (MDL) criterion is used for minimization of the stochastic complexity (SC) associated with the inferred population structure. To encode possible dependencies among the loci, we augment the classifier with a binary tree to enable a sparse factorization of the joint distribution of the multi-locus genotypes where the amount of dependence is determined by the level of complexity in the data when filtered through the MDL criterion. The mathematical model thus enables simultaneous learning of the unsupervised classification and the graphical structures that are needed for representing the possible dependencies. It will be explicitly shown that SC (per sample) is the sum of the length of the description of the samples within the classification and of the length of the description of the classification with respect to the classification model chosen. Some parts of our findings were earlier reported in a condensed form in a conference proceedings article [11]. We point out that the general notion of augmenting a classifier by a tree is due to Chow and Liu [12] and in a more extensive form due to [13] in a supervised context.

This article is organized as follows. In Section 2.1 we introduce a tree-based factorization of the joint distribution of the multilocus genotypes, in Section 2.2 the SC criteria for learning trees and unsupervised

classifications, respectively, are derived. The final sections thereafter discuss deterministic algorithms for learning the optimal population structure under the introduced framework and provide some concluding remarks.

2. Results and Discussion

2.1. Tree-based factorization of the joint distribution of multilocus genotypes

Assume the observed data consists in total of n observed genotypes over d biallelic marker loci from a haploid organism. Thus, each observation resides in the d -dimensional binary hypercube B^d and we let x denote its elements carrying d binary components, *i.e.*,

$$B^d := \left\{ x \mid x = (x_i)_{i=1}^d, x_i \in \{0, 1\} \right\}$$

Consider one latent pool in the population into which t of the total n samples are assigned in terms of an unsupervised classification into k disjoint classes. The classification itself will be made notationally explicit in a later section, whereas here we seek a probabilistic representation of the possible dependencies among the d allele frequencies of the individual loci in any particular pool. At one extreme, such representation should also allow for the simplest possible model where all loci are independent, thus encoding the situation where the data do not display evidence for dependence. Using the general Markov theory developed for graphical models of multivariate distributions, we may derive the sought probabilistic representation that enjoys these properties. We let

$$X^t = \{x^{(l)}\}_{l=1}^t \quad (1)$$

be a set of $x^{(l)} \in B^d$, which are considered as t independent realizations of the multilocus genotypes from a single latent pool with its specific (unknown) allele frequencies. Note that the data is here assumed complete, such that no components are missing in any $x^{(l)}$.

Let $\mathcal{G} = (V, E)$ be an acyclic graph with the set of nodes (or vertices) $V = \{1, \dots, d\}$ and the edges E . Each connected component of $\mathcal{G}$ is a *tree* and $\mathcal{G}$ may also be called a *forest*. We shall first consider forests that consist of one single tree.

If we choose a direction for the edges of $\mathcal{G}$, the node i in the directed edge (i, j) in E is said to be the parent of node j and j is called a child of the node i . The notations i and j are used interchangeably to denote any particular locus in the remainder of the text. A tree is characterized by the property that any two vertices are connected by a unique path. Hence the parent of a node is uniquely given. The root of a directed tree is the node lacking a parent. If $\mathcal{G}$ is a directed tree, then we designate by $\mathcal{G}^\sim$ the undirected version obtained from $\mathcal{G}$ by replacing the directed edges by undirected edges.

The *structure* or *topology* of the tree $\mathcal{G}$ (and effectively of $\mathcal{G}^\sim$) is thus given by

$$\Pi = (\Pi[1], \Pi[2], \dots, \Pi[d]),$$

where $\Pi[j]$ is the parent of the node j , such that $\Pi(1) = \emptyset$ (the empty set). We suppose here that the nodes are ordered so that 1 is the root of the tree and that $\Pi(i) < i$ for all i (topological order).

For each node we assign a binary random variable X_i corresponding to the i th locus, where the possible alleles are labeled by $x_i \in \{0, 1\}$. Each edge (j, i) in E (directed or not) is a statement of

dependence between X_j and X_i , which implies that the joint allele frequencies of the locus combination j, i do not factorize into the product of the marginal allele frequencies. In contrary, the absence of an edge indicates lack of direct dependence between the corresponding loci.

We assume that the joint distribution of $(X_1, \dots, X_d)$ is factorized along $\mathcal{G}$ in the sense that

$$P(X_1 = x_1, \dots, X_d = x_d) = \prod_{i=1}^d P(X_i = x_i | X_{\Pi[i]} = x_{\Pi[i]}) \quad (2)$$

In addition to the structure Π we thus need to assign for each node the table of conditional probabilities $P(X_i = x_i | X_{\Pi[i]} = x_{\Pi[i]})$ (allele frequencies) in order to fully specify a tree dependent joint distribution.

Simple Markovian process based dependence representations of linkage between loci have earlier been introduced in [8] and [14], where it was assumed that the linkage map of the loci is available *a priori* and it was explicitly incorporated in the models. The tree and forest based factorizations of the joint allele probabilities defined below also utilize simple Markovian structures along the node ordering as any locus will have only a single or no parents. The essential difference is then that the linkage map of the loci is not assumed known *a priori*, but is implicitly represented by the node ordering along the tree or forest, which is learned from the observed data. In situations where the chromosome at which a locus is positioned is known for the loci, it is straightforward to impose the information to our framework by learning the trees (forests) separately for each subset of loci positioned within different chromosomes. In order not to burden the notation excessively, we abstain from an explicit representation of this possibility.

For notational convenience we shall in the sequel express the joint distributions $P(X_1 = x_1, \dots, X_d = x_d)$ and other probability distributions also by omitting the random variables but including the structure of the tree as

$$P(x_1, \dots, x_d | \Pi) = \prod_{i=1}^d P(x_i | x_{\Pi[i]}).$$

A directed tree $\mathcal{G} = (V, E)$ equipped with tree dependent probability distribution (2) factorized along $\mathcal{G} = (V, E)$ is the model introduced in [12] and it is also called a dependence tree. The tree, the binary random variables $X_1, \dots, X_d$, and the probability distribution (2) constitute a special *Bayesian network* [15,16]. However, Meil and Jordan [17] argue that a mixture of trees, which will be introduced below using the probability distributions (2) as components, is not to be regarded as a Bayesian network.

The undirected graph $\mathcal{G}^\sim$ is also known as the *Markov tree* of the probability distribution $P(x_1, \dots, x_d | \Pi)$ [18]. A tree dependent distribution factorized along $\mathcal{G}^\sim$ with $P(x_1, \dots, x_d | \Pi) > 0$ enjoys the global Markov property, and hence vertex separation implies conditional independence [16]. Therefore $\mathcal{G}^\sim$ is a perfect representation of $P(x_1, \dots, x_d | \Pi)$ [19].

An example of a Chow-Liu dependence tree in the sense of above presentation for seven loci is given by

$$(\Pi[1], \Pi[2], \Pi[3], \Pi[4], \Pi[5], \Pi[6], \Pi[7]) = (\emptyset, 1, 2, 1, 4, 5, 3),$$

which leads to the factorization

$$P(x_1) P(x_2|x_1) P(x_3|x_2) P(x_4|x_1) P(x_5|x_4) P(x_6|x_5) P(x_7|x_3),$$

of the joint distribution of the alleles within the considered latent pool. An illustration of a disconnected dependence tree for seven nodes is given by

$$(\Pi[1], \Pi[2], \Pi[3], \Pi[4], \Pi[5], \Pi[6], \Pi[7]) = (\emptyset, 1, \emptyset, \emptyset, 4, 4, 5),$$

which now leads to the factorization

$$P(x_1) P(x_2|x_1) P(x_3) P(x_4) P(x_5|x_4) P(x_6|x_4) P(x_7|x_5). \quad (3)$$

Bayesian networks can be equivalent in the sense that they imply the same set of independencies between the variables. Or, they have the same underlying undirected graph but might disagree on the direction of some of the edges. One cannot distinguish between equivalent graphs using observations of the variables of the graph. Two rooted trees $\mathcal{G}_1$ and $\mathcal{G}_2$ are equivalent, if they have the same underlying undirected graph $\mathcal{G}^\sim$. The characterizations of equivalence are recapitulated and proved in [20,21]. In the current work the dependencies between loci are unambiguously represented by the undirected graph $\mathcal{G}^\sim$.

2.2. Chow expansion of the joint distribution of multilocus genotypes

Since the joint distribution factorized along a forest of dependence trees will, in view of (3), even in general consist of factors having the same structure as for the joint distribution factorized according to one single dependence tree, we restrict the attention to the tree dependent probability distribution

$$P(x) = P(x_1) \cdot P(x_2|x_{\Pi(2)}) \dots P(x_{d-1}|x_{\Pi(d-1)}) \cdot P(x_d|x_{\Pi(d)}). \quad (4)$$

The factors $P(x_i|x_{\Pi(i)})$ can be written as

$$P(x_i|x_{\Pi(i)}) = \left(\theta_i^{x_i} (1 - \theta_i)^{(1-x_i)} \right)^{x_{\Pi(i)}} \cdot \left(\phi_i^{x_i} (1 - \phi_i)^{(1-x_i)} \right)^{1-x_{\Pi(i)}}, \quad (5)$$

where for $i = 2, \dots, d$

$$\theta_i = P(x_i = 1 | x_{\Pi(i)} = 1), \quad (6)$$

and

$$\phi_i = P(x_i = 1 | x_{\Pi(i)} = 0), \quad (7)$$

and

$$P(x_1) = \theta_1^{x_1} (1 - \theta_1)^{(1-x_1)}.$$

Recalling from the previous section definition of the data that was assigned to a particular latent pool, we obtain the joint probability of the t d -dimensional samples from (4) and (5) as

$$\prod_{l=1}^t P(x^{(l)}) = \theta_1^{n_1} (1 - \theta_1)^{t-n_1} \prod_{i=2}^d \theta_i^{n_i(1,1)} (1 - \theta_i)^{n_i(0,1)} \cdot \phi_i^{n_i(1,0)} (1 - \phi_i)^{n_i(0,0)}, \quad (8)$$

where, as all $x^{(l)}$ are realizations under the same dependence tree,

$$n_i(1, 1) = \sum_{l=1}^t x_i^{(l)} x_{\Pi(i)}^{(l)}, \quad n_i(1, 0) = \sum_{l=1}^t x_i^{(l)} \left(1 - x_{\Pi(i)}^{(l)} \right),$$

$$n_i(0, 1) = \sum_{l=1}^t \left(1 - x_i^{(l)}\right) x_{\Pi(i)}^{(l)}, \quad n_i(0, 0) = \sum_{l=1}^t \left(1 - x_i^{(l)}\right) \left(1 - x_{\Pi(i)}^{(l)}\right),$$

for $i = 2, \dots, d$, and

$$n_1 = \sum_{l=1}^t x_1^{(l)}.$$

Obviously, $n_i(1, 1)$ counts the number of times we have simultaneously $x_i^{(l)} = 1$ and $x_{\Pi(i)}^{(l)} = 1$ in $X^t = \{x^{(l)}\}_{l=1}^t$, and the interpretations of the remaining corresponding quantities are also obvious. We call (8) the *(first order) Chow expansion of the joint likelihood of a Markov tree* [22].

Prior predictive data distributions under Chow expansion

Since the Chow expansion derived above involves a considerable amount of unknown allele frequency parameters and different combinations of trees and unsupervised classifications will be associated with varying numbers of such parameters, it is necessary to handle them in an appropriate manner from the statistical perspective, to ensure coherent learning of the population structure. Here we use the *marginal likelihood* of the tree topology $P(X^t|\Pi)$, which can equivalently be considered as the *prior predictive distribution* of X^t , to ensure consistent learning under the family of the considered probability models.

The joint likelihood of the tree network in (5) is now represented by the parametrization

$$P_{\underline{\theta}, \underline{\phi}}(x) = \prod_{i=1}^d P_{\theta_i, \phi_i}(x_i | x_{\Pi(i)}),$$

where from (6) and (7)

$$\underline{\theta} = (\theta_1, \dots, \theta_d), \quad \underline{\phi} = (\phi_2, \dots, \phi_d).$$

Given that Θ and Φ denote two copies of the d -fold and $d - 1$ -fold product of the unit interval, respectively, then $\underline{\theta} \in \Theta$ and $\underline{\phi} \in \Phi$.

Given a prior probability density $g(\underline{\theta}, \underline{\phi})$ on $\Theta \times \Phi$, we obtain

$$P(X^t|\Pi) = \int_{\Theta} \int_{\Phi} \prod_{l=1}^t P_{\underline{\theta}, \underline{\phi}}(x^{(l)}) g(\underline{\theta}, \underline{\phi}) d\underline{\theta} d\underline{\phi}. \quad (9)$$

This is known in statistics as a prior predictive distribution. We choose the prior $g(\underline{\theta}, \underline{\phi})$ by *local meta independence* of parameters, [15,20,21], such that

$$g(\underline{\theta}, \underline{\phi}) = \prod_{i=1}^d h(\theta_i) \prod_{i=2}^d z(\phi_i). \quad (10)$$

An explicit expression for the above prior predictive data distribution is derived in Appendix.

2.3. Stochastic complexity and learning of classifications and tree structures

Given the above results we can now derive an expression of the *stochastic complexity* for any Chow expansion of joint allele frequencies in a latent pool in a sense made precise in [23,24]. This result will later be invoked to the task of unsupervised classification, such that our learning approach will aim at

minimizing the SC over the space of all possible combinations of latent pools and their associated Chow expansions. By using the result in (9), we obtain the criterion for minimizing the SC as the expression

$$\begin{aligned}
 -\log P(X^t|\Pi) &= -\log I_1 - \log I_2 - \log I_3 \\
 &= \log \frac{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)} + \log \frac{\Gamma(t + \alpha_1 + \alpha_2)}{\Gamma(n_1 + \alpha_1) \cdot \Gamma(t - n_1 + \alpha_2)} \\
 &\quad + \sum_{i=2}^d \log \frac{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)} \frac{\Gamma(n_i(1, 1) + n_i(0, 1) + \alpha_1 + \alpha_2)}{\Gamma(n_i(1, 1) + \alpha_1) \cdot \Gamma(n_i(0, 1) + \alpha_2)} \\
 &\quad + \sum_{i=2}^d \log \frac{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)} \frac{\Gamma(n_i(1, 0) + n_i(0, 0) + \alpha_1 + \alpha_2)}{\Gamma(n_i(1, 0) + \alpha_1) \cdot \Gamma(n_i(0, 0) + \alpha_2)},
 \end{aligned} \tag{11}$$

where the hyperparameters are chosen as $\alpha_1 = \alpha_2 = 1/2$, which defines a product of respective Jeffreys' priors in (10). In order to minimize the log probability $-\log P(X^t|\Pi)$ we derive an asymptotic expansion of the above expression. This expansion is instructive both for algorithmic purposes and for gaining explicit illustration of the mechanics of SC minimization.

Asymptotic expansion of the stochastic complexity for a Chow expansion

The formulae derived in Appendix establish the following basic result. If the likelihood of the tree topology $P(X^t|\Pi)$ for a dependence tree is evaluated assuming both local meta independence and Jeffreys' prior, which equals the Beta distribution $\text{Be}(1/2, 1/2)$, for each of the parameters, then

$$\begin{aligned}
 -\log P(X^t|\Pi) &= \log \frac{t!}{\Gamma(n_1 + 1/2) \cdot \Gamma(t - n_1 + 1/2)} \\
 &\quad + t \cdot \sum_{i=2}^d h(n_i(1)/t) - t \cdot \sum_{i=2}^d I_{i,\Pi(i)} \\
 &\quad + \frac{1}{2} \sum_{i=2}^d \log(n_{\Pi(i)}(1)) + \log(n_{\Pi(i)}(0)) + C,
 \end{aligned} \tag{12}$$

where $I_{i,\Pi(i)}$ is given in (50) and C is bounded in t .

Since the learning of the tree network structure is achieved by minimizing $-\log P(X^t|\Pi)$ as a function of the structure Π , in view of (12) we see that this corresponds to minimizing

$$-t \cdot \sum_{i=2}^d I_{i,\Pi(i)} + \frac{1}{2} \sum_{i=2}^d [\log(n_{\Pi(i)}(1)) + \log(n_{\Pi(i)}(0))],$$

since all other terms are independent of the network structure. This is additionally equivalent to maximization of

$$\sum_{i=2}^d I_{i,\Pi(i)} - \frac{1}{2} \cdot \left[\frac{1}{t} \sum_{i=2}^d (\log(n_{\Pi(i)}(1)) + \log(n_{\Pi(i)}(0))) \right]. \tag{13}$$

The stochastic complexity for an unsupervised classification under Chow expansion

We now consider derivation of SC for unsupervised classifications of the n samples into k disjoint classes interpreted to represent latent pools in the population, each associated with distinct allele

frequencies. In terms of data, an unsupervised classification refers to a subdivision of the observed multilocus genotypes into the sets $\{X^{t_c}, c = 1, \dots, k\}$. There are various ways of representing such subdivisions, here we define them in terms of class membership functions

$$u_c^{(l)} := \begin{cases} 1 & \text{if } x^{(l)} \in c \\ 0 & \text{otherwise,} \end{cases}$$

and incorporate these in the matrix

$$U^n = \{u_c^{(l)}\}_{l=1, c=1}^{n, k}.$$

Hence

$$t_c = \sum_{l=1}^n u_c^{(l)} \quad (14)$$

is the number of binary vectors assigned to class $c = 1, \dots, k$.

Let next $\lambda = \{\lambda_c\}_{c=1}^k$ be a discrete probability distribution $\sum_{c=1}^k \lambda_c = 1, \lambda_c \geq 0$, which we use as the prevalence of $u_c^{(l)}$ so that

$$\lambda_c = P(u_c^{(l)} = 1), \quad l = 1, \dots, n. \quad (15)$$

This can be interpreted as the probability of the event that $x^{(l)}$ is sampled from the latent pool indexed by c . Then we introduce

$$P_{\underline{\theta}_c, \underline{\phi}_c}(x | \Pi_c) = \prod_{i=1}^d P_{\theta_{ic}, \phi_{ic}}(x_i | x_{\Pi_c(i)}) \quad (16)$$

as the conditional probability of x within the latent pool c . Thus, each of the classes is equipped with a dependence tree or possibly a forest of dependence trees designated by Π_c . We can now write using (15) and (16)

$$P(x) = \sum_{c=1}^k \lambda_c P_{\underline{\theta}_c, \underline{\phi}_c}(x | \Pi_c), \quad (17)$$

which is formally a *mixture of trees* in the sense of Meil and Jordan [17]. A difference in the present setting is that each Π_c may also be a forest. Figure 1 provides an illustration of the joint structure of the probability model for a simple example with $d = 5$ and $k = 2$.

In view of the preceding we have the *complete integrated likelihood*

$$P((X^n, U^n) | \underline{\Pi}) = \int_{\Lambda} \int_{\Theta} \int_{\Phi} \prod_{l=1}^n \prod_{c=1}^k \left[P_{\underline{\theta}_c, \underline{\phi}_c}(x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}} g(\underline{\theta}_c, \underline{\phi}_c) \psi(\lambda) d\underline{\theta}_c d\underline{\phi}_c d\lambda,$$

where we have $\underline{\Pi}$ denoting the collection of the trees (forests) $\{\Pi_c, c = 1, \dots, k\}$, and further

$$\underline{\theta}_c = (\theta_{1c}, \dots, \theta_{dc}), \quad \underline{\phi}_c = (\phi_{2c}, \dots, \phi_{dc})$$

and Θ and Φ are the appropriate spaces for $\underline{\theta}_1, \dots, \underline{\theta}_k$ and $\underline{\phi}_1, \dots, \underline{\phi}_k$ to assume their values in. The space Λ is equal to $\{\lambda | \sum_{c=1}^k \lambda_c = 1, \lambda_c \geq 0\}$ and $\psi(\lambda)$ is a probability density on Λ . The above integral factorizes as

$$P(X^n, U^n | \underline{\Pi}) = Q^*(U^n) \cdot P^*(X^n | U^n, \underline{\Pi}), \quad (18)$$

Figure 1. Graphical representation of the dependence structure for an unsupervised classification model augmented by Chow-Liu trees. Here $d = 5$ and $k = 2$ and the unbroken arrow lines correspond to dependence between the stochastic nodes and the dashed arrows correspond to the dependence of the root nodes on the classification variable λ , which is connected to the trees by a random switch (represented by the curved arrow) according to the probabilities in λ .



where the auxiliary notations are defined as

$$Q^*(U^n) = \int_{\Lambda} \prod_{c=1}^k \lambda_c^{t_c} \psi(\lambda) d\lambda, \quad (19)$$

and

$$P^*(X^n | U^n, \Pi) = \prod_{c=1}^k \int_{\Theta_c} \int_{\Phi_c} \prod_{l=1}^n \left[P_{\theta_c, \phi_c}(x^{(l)} | \Pi_c) \right]^{u_c^{(l)}} g(\theta_c, \phi_c) d\theta_c d\phi_c.$$

With regard to the last integral above the situation is analogous to (8) so that

$$\prod_{l=1}^n \left[P_{\theta_c, \phi_c}(x^{(l)} | \Pi_c) \right]^{u_c^{(l)}} = K_1 \prod_{i=2}^d \theta_{ic}^{n_{ic}(1,1)} (1 - \theta_{ic})^{n_{ic}(0,1)} \cdot \phi_{ic}^{n_{ic}(1,0)} (1 - \phi_{ic})^{n_{ic}(0,0)}, \quad (20)$$

where θ_{ic} is defined as

$$\theta_{ic} = P(x_i = 1 | x_{\Pi_c(i)} = 1)$$

and analogously for ϕ_{ic} and

$$K_1 = \theta_{1c}^{n_{1c}} (1 - \theta_{1c})^{t_c - n_{1c}}.$$

As all $x^{(l)}$ with the label c follow the factorization of the multivariate distribution for the same dependence tree Π_c ,

$$n_{ic}(1, 1) = \sum_{l: u_c^l = 1} x_i^{(l)} x_{\Pi_c(i)}^{(l)}, n_{ic}(1, 0) = \sum_{l: u_c^l = 1} x_i^{(l)} (1 - x_{\Pi_c(i)}^{(l)}), \quad (21)$$

$$n_{ic}(0, 1) = \sum_{l: u_c^l = 1} (1 - x_i^{(l)}) x_{\Pi_c(i)}^{(l)}, n_{ic}(0, 0) = \sum_{l: u_c^l = 1} (1 - x_i^{(l)}) (1 - x_{\Pi_c(i)}^{(l)}), \quad (22)$$

for $i = 2, \dots, d$ and

$$n_{1c} = \sum_{l: u_c^l = 1} x_1^{(l)}.$$

and t_c is given in (14). Here $n_{ic}(1, 1)$ counts the number of times the pair $(x_i^{(l)} = 1, x_{\Pi_c(i)}^{(l)} = 1)$ occurs in the data set simultaneously with $u_c^{(l)} = 1$.

Hence, under the above assumptions the same computations that were used in the preceding section entail the formula

$$\begin{aligned} -\log P^*(X^n|U^n, \underline{\Pi}) &= -\sum_{c=1}^k \log P(X^n|\Pi_c) \\ &= k \log \pi + \sum_{c=1}^k \log \frac{t_c!}{\Gamma(n_{1c} + 1/2) \cdot \Gamma(t_c - n_{1c} + 1/2)} + \\ &\quad + 2 \cdot k \cdot (d-1) \cdot \log \pi \\ &\quad + \sum_{c=1}^k \sum_{i=2}^d \log \frac{(n_{ic}(1, 1) + n_{ic}(0, 1))!}{\Gamma(n_{ic}(1, 1) + 1/2) \cdot \Gamma(n_{ic}(0, 1) + 1/2)} \\ &\quad + \sum_{c=1}^k \sum_{i=2}^d \log \frac{(n_{ic}(1, 0) + n_{ic}(0, 0))!}{\Gamma(n_{ic}(1, 0) + 1/2) \cdot \Gamma(n_{ic}(0, 0) + 1/2)}, \end{aligned} \quad (23)$$

where we have introduced the notation $-\log P(X^n|\Pi_c)$ for the expression in (11) when the class index c is inserted in all the appropriate places. The forest of Chow-Liu trees is merely implicit in the right hand side of this expression, but is, of course, required for computation and definition of the statistics of the locus pairs.

The integral $\int_{\Lambda} \prod_{c=1}^k \lambda_c^{t_c} \psi(\lambda) d\lambda$ is explicitly evaluated taking $\psi(\lambda)$ as the Dirichlet density, see [26] for a rationale, which means that

$$\psi(\lambda_1, \dots, \lambda_k) = \begin{cases} \frac{\Gamma(\sum_{c=1}^k \alpha_c)}{\prod_{c=1}^k \Gamma(\alpha_c)} \prod_{c=1}^k \lambda_c^{\alpha_c-1}, & \text{if } \lambda_1, \dots, \lambda_k \in \Lambda \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

Then in (19)

$$\begin{aligned} Q^*(U^n) &= \int_{\Lambda} \prod_{c=1}^k \lambda_c^{t_c} \psi(\lambda) d\lambda = \frac{\Gamma(\sum_{c=1}^k \alpha_c)}{\prod_{c=1}^k \Gamma(\alpha_c)} \int_{\Lambda} \prod_{c=1}^k \lambda_c^{t_c + \alpha_c - 1} d\lambda = \\ &= \frac{\Gamma(\sum_{c=1}^k \alpha_c)}{\prod_{c=1}^k \Gamma(\alpha_c)} \cdot \frac{\prod_{c=1}^k \Gamma(t_c + \alpha_c)}{\Gamma(\sum_{c=1}^k t_c + \sum_{c=1}^k \alpha_c)}. \end{aligned}$$

If we choose the Jeffreys' prior, which corresponds to $\alpha_c = 1/2$ (for this result and the computations required, c.f., [24] or [27] p. 218, for all c , we obtain

$$\begin{aligned} -\log Q^*(U^n) &= \frac{k}{2} \log \pi/2 - \log \Gamma(k/2) \\ &\quad + \log \Gamma(n + k/2) - \sum_{c=1}^k \log \Gamma(t_c + 1/2). \end{aligned} \quad (25)$$

We have thus established that SC (per item) is the sum of the length of the description of the items within the classification and of the length of the description of the classification with respect to the classification model chosen, *i.e.*,

$$SC := -\frac{1}{n} \log P^*(X^n|U^n, \underline{\Pi}) + \left(-\frac{1}{n} \log Q^*(U^n) \right), \quad (26)$$

respectively, with detailed expressions from (23) and (25). This formula can be used to evaluate the complexity of any classification U^n of data X^n , irrespective of the way it has been arrived at.

2.4. Algorithms for learning unsupervised classifications and Chow expansions

Deterministic algorithm for learning Chow expansions

Chow and Liu [12] established an algorithm for maximization of $\sum_{i=2}^d I_{i,\Pi(i)}$ as a function of the tree structure. Suzuki [28] has extended the Chow-Liu algorithm by adding to $\sum_{i=2}^d I_{i,\Pi(i)}$ terms penalizing tree networks that are too complex. We shall next present a variant of these two algorithms for *any single class c* containing t_c samples.

The procedure for constructing disconnected tree networks consists of the following steps:

A1. Compute the numbers

$$IP_{i,j} = \sum_{u=0}^1 \sum_{v=0}^1 \hat{P}_{i,j}(u,v) \log \frac{\hat{P}_{i,j}(u,v)}{\hat{P}_i(u) \cdot \hat{P}_j(v)} - \frac{1}{2} \cdot \frac{1}{t} [\log(n_c(1)) + \log(n_c(0))] \quad (27)$$

for the $d \cdot (d-1)$ different pairs of indices, where $\hat{P}_{i,j}(u,v)$, $\hat{P}_i(u)$, $\hat{P}_j(v)$ refer to maximum likelihood estimates of the corresponding probabilities (see formula (50) in Appendix).

A2. Construct a complete undirected graph with the binary variables as nodes.

A3. Construct a maximum weighted spanning tree with the extra condition that an edge is in the tree only if $IP_{i,j} > 0$.

A4. Make the maximum weighted spanning tree directed by choosing a root variable and setting the direction of all edges to be outward from the root.

There are several algorithms for constructing a maximum weighted spanning tree in step **A3**, when the condition for permitting disconnected graphs is not imposed. The most time honoured algorithm for the task is the *Borůvka-Choquet-Kruskal algorithm* [29].

Having completed the steps **A1-A4** we have a tree structure

$$\hat{\Pi}_c = \left\{ \hat{\Pi}_c^{(s)} \right\}_{s=1}^r, \quad (28)$$

where each $\hat{\Pi}_c^{(s)}$ corresponds to Chow-Liu tree with a subset of the nodes $\{1, \dots, d\}$ and its distinct and separate root. For a disconnected tree we then have the stochastic complexity

$$-\log P(X^{t_c} | \hat{\Pi}_c) = -\sum_{s=1}^r \log P(X^{t_c} | \hat{\Pi}_c^{(s)}), \quad (29)$$

where each term in the right hand side is of the form

$$\begin{aligned}
 -\log P\left(X^{t_c}|\hat{\Pi}_c^{(s)}\right) &= \log \pi + \log \frac{t_c^{(s)}!}{\Gamma\left(n_1^{(s)}+1/2\right) \cdot \Gamma\left(t_c^{(s)}-n_1^{(s)}+1/2\right)} \\
 &+ (d^{(s)}-1) \log \pi \\
 &+ \sum_{i=2}^{d^{(s)}} \log \frac{\Gamma\left(n_{\Pi(i)}^{(s)}(1)+1\right)}{\Gamma\left(n_i^{(s)}(1,1)+1/2\right) \cdot \Gamma\left(n_i^{(s)}(0,1)+1/2\right)} \\
 &+ (d^{(s)}-1) \log \pi \\
 &+ \sum_{i=2}^{d^{(s)}} \log \frac{\Gamma\left(n_{\Pi(i)}^{(s)}(0)+1\right)}{\Gamma\left(n_i^{(s)}(1,0)+1/2\right) \cdot \Gamma\left(n_i^{(s)}(0,0)+1/2\right)},
 \end{aligned} \tag{30}$$

with obvious definitions of the quantities involved.

The case $d^{(s)} = 1$ corresponds to a node that is not connected to any other node, and thereby only the first term in the right hand side is needed in (30).

Deterministic algorithm for learning unsupervised classification augmented by Chow expansions

Recall the logarithm of the marginal likelihood of the data as a function of an instance of an unsupervised classification and the set of Chow expansions,

$$L\left((X^n, U^n) | \underline{\Pi}\right) = \log P\left((X^n, U^n) | \underline{\Pi}\right) \tag{31}$$

Below we show how an algorithm for computing $\max_{U^n, \underline{\Pi}} \frac{1}{n} L\left((X^n, U^n) | \underline{\Pi}\right)$ can be formulated in terms of a maximum likelihood estimation procedure. Under the previously stated assumptions, as n grows to infinity, we have for fixed X^n and k the expression

$$\begin{aligned}
 \max_{U^n, \underline{\Pi}} \frac{1}{n} L\left((X^n, U^n) | \underline{\Pi}\right) &= \max_{U^n, \underline{\Pi}, \Theta, \lambda} \frac{1}{n} \sum_{l=1}^n \sum_{c=1}^k \left[u_c^{(l)} \log P_{\underline{\theta}_c, \underline{\phi}_c} \left(x^{(l)} | \Pi_c\right) + u_c^{(l)} \log \lambda_c \right] \\
 &- \frac{1}{2} k \cdot (2d) \frac{\log n}{n} + R,
 \end{aligned} \tag{32}$$

where R is bounded in n for fixed X^n and k . By stating that X^n is fixed as $n \rightarrow \infty$, we mean that when $x^{(n+1)}$ is added, the preceding $x^{(l)}$ in X^n , $l \leq t$ are not changed. We shall merely show that the desired expansion is another way of writing (12) above, when terms corresponding to the similar expansion of $-\log Q^*(U^n)$ in (25) are added.

To evaluate the above expression more explicitly, we start by considering in the right hand side of (32) the maximization of the likelihood

$$\max_{U^n, \underline{\Pi}, \Theta, \lambda} \prod_{l=1}^n \prod_{c=1}^k \left[P_{\underline{\theta}_c, \underline{\phi}_c} \left(x^{(l)} | \Pi_c\right) \lambda_c \right]^{u_c^{(l)}},$$

which is equivalent to the maximization of

$$\max_{U^n, \underline{\Pi}, \Theta, \lambda} \frac{1}{n} \sum_{l=1}^n \sum_{c=1}^k \left[u_c^{(l)} \log P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) + u_c^{(l)} \log \lambda_c \right].$$

We first maximize with respect λ to obtain

$$\max_{U^n, \underline{\Pi}} \frac{1}{n} \sum_{l=1}^n \sum_{c=1}^k \left[u_c^{(l)} \log P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) \right] + \sum_{c=1}^k \hat{\lambda}_c \log \hat{\lambda}_c, \quad (33)$$

where $\hat{\lambda}_c$ is

$$\hat{\lambda}_c = \frac{\sum_{l=1}^n u_c^{(l)}}{n} = \frac{t_c}{n}. \quad (34)$$

But then the argument following (47) shows that

$$\begin{aligned} \frac{1}{n} \sum_{c=1}^k \sum_{l=1}^n u_c^{(l)} \log P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) &= \frac{1}{n} \sum_{c=1}^k \log K_{1c} \\ &+ \frac{1}{n} \sum_{c=1}^k \log \frac{t_c!}{\Gamma(n_{1c} + 1/2) \cdot \Gamma(t_c - n_{1c} + 1/2)} \\ &- \sum_{c=1}^k \frac{t_c}{n} \sum_{i=2}^d h(n_{ic}(1)/n) + \sum_{c=1}^k \frac{t_c}{n} \sum_{i=2}^d I_{i, \Pi_c(i)}. \end{aligned} \quad (35)$$

In other words we are going to maximize

$$\frac{1}{n} \sum_{c=1}^k \sum_{l=1}^n u_c^{(l)} \log P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c)$$

by maximization of

$$\sum_{c=1}^k \frac{t_c}{n} \sum_{i=2}^d I_{i, \Pi_c(i)},$$

as a function of $\underline{\Pi}$, which we can do for each class c separately using the Chow-Liu algorithm. Having found the optimum tree topology, we have also obtained the maximum likelihood estimates $\hat{\underline{\theta}}_c, \hat{\underline{\phi}}_c$ for each $c, c = 1, \dots, k$.

However, we have already in (12) the additional terms $\frac{1}{2} \sum_{i=2}^d \log(n_{\Pi_c(i)}(1)) + \log(n_{\Pi_c(i)}(0))$, which can be subsumed in $\frac{1}{2} k \cdot (2d) \frac{\log n}{n}$.

Clearly we can use Stirling's formula in a similar way as in (44) to expand $-\log Q^*(U^n)$ in (25) so as to obtain

$$-\frac{1}{n} \log Q^*(U^n) \approx - \sum_{c=1}^k \hat{\lambda}_c \log \hat{\lambda}_c - \frac{k-1}{2} \frac{\log n}{n}.$$

The general asymptotic expansion result due Schwartz [30], as in [31], c.f. [32] and [33] chapter 5, provides a similar kind of an expansion, however, without making comparable assumptions about the prior densities. This special application is based on the fact that $\left[P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}}$ belongs to the

exponential family of distributions with the convex parameter sets. In general, study of the asymptotics of approximate Bayesian model selection under implicit priors in the presence of hidden states like class variables is very challenging; for some recent significant progress see [33,34].

We shall next consider an algorithm for unsupervised classification of X^n , i.e., for finding U^n that maximizes $L((X^n, U^n) | \Pi)$ for a given value of k , using the expansion in (31). A discussion of this kind of expansions in clustering theory is found in [35]. We define first the rules of identification to be used in the algorithm.

A *tree augmented supervised Bayesian classifier* [13] is based on the following rule of identification first suggested by Chow and Liu [12]. An x in B^d is assigned to (identified with) class c^* , if

$$c_* = \arg \max_{1 \leq c \leq k} P_{\hat{\theta}_c, \hat{\phi}_c}(x | \Pi_c) \hat{\lambda}_c, \quad (36)$$

where $\hat{\lambda}_c$ is given in (34) and

$$P_{\hat{\theta}_c, \hat{\phi}_c}(x | \Pi_c) = \prod_{i=1}^d P_{\hat{\theta}_{ic}, \hat{\phi}_{ic}}(x_i | x_{\Pi(i)}). \quad (37)$$

In order to simplify the required notation, we drop here the superscript from the data matrix and set $U^n = U$. The findings in the preceding subsection show that we can maximize

$$\max_{U, \Pi, \Theta, \lambda} \prod_{l=1}^n \prod_{c=1}^k \left[P_{\theta_c, \phi_c}(x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}}$$

using the following alternating algorithm:

B1. Fix k , set $w = 0$ and store an arbitrary (random) $U_{(w)}$.

B2. Find the structure $\hat{\Pi}_{(w)}$ maximizing

$$\sum_{c=1}^k \frac{t_c}{n} \sum_{i=2}^d I_{i, \Pi_c(i)} - \frac{1}{2} k \cdot (2d) \frac{\log n}{n}$$

using the previously defined Chow-Liu-Suzuki algorithm (A1-A4).

B3. For $U_{(w)}$ and $\hat{\Pi}_{(w)}$ compute the maximum likelihood estimates $\hat{\Theta}_{(w)}$ and $\hat{\lambda}_{(w)}$.

B4. Given $\hat{\Theta}_{(w)}$, $\hat{\lambda}_{(w)}$, and $\hat{\Pi}_{(w)}$ determine $U_{(w+1)} = \left\{ (u_c^{(l)})_{(w+1)} \right\}_{c,l=1}^{n,k}$ using

$$(u_c^{(l)})_{(w+1)} = \begin{cases} 1 & \text{if } c_*^{(l)} = c \\ 0 & \text{otherwise,} \end{cases} \quad (38)$$

where

$$c_*^{(l)} = \arg \max_{1 \leq c \leq k} P_{\hat{\theta}_c, \hat{\phi}_c}(x^{(l)} | \Pi_c) \hat{\lambda}_c.$$

B5. If $U_{(w+1)} = U_{(w)}$, then stop, otherwise set $w = w + 1$ and go to **B2**.

It can be proved in the same way as in [31] that this algorithm will re-enter step **B2** only a finite number of times and, after having stopped, will have found a local minimum of

$$\prod_{l=1}^n \prod_{c=1}^k \left[P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}}$$

as a function of $U, \underline{\Pi}, \Theta, \lambda$. This is easily seen, since each step of the algorithm above increases the value of the likelihood function, since only non-negative terms are added in step **B4**.

Due to local meta independence and the other assumptions the class wise components in $\hat{\Theta}_{(w)}$, $\hat{\lambda}_{(w)}$, and $\hat{\Pi}_{(w)}$ are estimated using those items in $U_{(w)}$ assigned to the class c , respectively, at this step of the algorithm.

The estimation procedure above, *i.e.*, finding $\hat{U}, \hat{\underline{\Pi}}, \hat{\Theta}, \hat{\lambda}$ such that

$$(\hat{U}, \hat{\underline{\Pi}}, \hat{\Theta}, \hat{\lambda}) = \arg \max_{U, \underline{\Pi}, \Theta, \lambda} \prod_{l=1}^n \prod_{c=1}^k \left[P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}}$$

is an example of what is known as the *classification maximum likelihood estimate*. The procedure has been proved to yield biased estimates of the parameters of the probability distribution [35]. In addition, the family of distributions dealt with here is not identifiable [31]. However, despite of this the classification performance need not be impaired in practice, provided that the underlying classes are represented by a wealth of samples.

Finally, in order to establish k , the number of classes, from the data X^n it is possible to proceed by executing the above algorithm for all feasible values of k and choose $\hat{k}$ and the corresponding $\hat{U}$ such that

$$-\frac{1}{n} \log P^* (X^n | \hat{U}^n, \hat{\underline{\Pi}}) + \left(-\frac{1}{n} \log Q^* (\hat{U}^n) \right),$$

is maximal.

In view of the fact that we are actually dealing with an exponential family in $\left[P_{\underline{\theta}_c, \underline{\phi}_c} (x^{(l)} | \Pi_c) \lambda_c \right]^{u_c^{(l)}}$, we note the consistency result in [36], which establishes the fact that maximizing

$$L((X^n, U^n) | \underline{\Pi})$$

as a function of k will produce a consistent estimate of the model and k .

2.5. Discussion

As the success of statistical mixture models applied to inferring population structures neatly demonstrates, general probabilistic machine learning theory contains many contributions that have potentially fruitful applications in diverse areas of scientific inquiry. Molecular biology in general is both an attractive target to applications of generic probabilistic machine learning tools as well as a source of inspiration for theoretical research on such methods, given the rich variety of biological problems that necessitate the use of advanced computational and statistical methods to arrive at meaningful solutions.

From a theoretical perspective it is fairly intuitive to attempt to represent possible dependencies between marker loci using the relatively sparse model structures the general theory of probabilistic

graphical models and networks has to offer. An introduction to the general theory can be found in [16]. It is worth noting that while our machine learning formulation does not include representation of linkage distances in terms of explicit parameters for that purpose, conditional distributions of alleles defined in terms of the tree factorization can still flexibly represent a wide range of dependencies from near independence to complete linkage between loci. Consequently, the evolutionary time scale related to the linkage patterns remains implicit in our formulation, and it is therefore dependent on the characteristics of a particular data set that is investigated.

We have intentionally abstained from considering explicitly any ancestral relationships of the observed samples in terms of a graph (or a tree); either at the level of individual samples or inferred latent pools of them. Such graphs would obviously increase the biological realism our approach when incorporated in the dependence model, however, as the level of computational complexity associated with our machine learning method is already very high, explicit models of ancestral relationships would likely render it practically inapplicable to data sets harboring large numbers of samples and latent pools.

In a typical theoretical formulation, probabilistic classification is based on ranking of the posterior probabilities of classes given an observed feature vector. This is in fact the optimal rule of identification if the “true” description of the data is used to compute the posterior probabilities. Wong and Poon [37] claimed that the tree-aided classifier of Chow and Liu minimizes an upper bound on the Bayes error rate, if the true distribution is approximated by (a mixture of) tree dependent distributions. However, it was later shown in [38] that the result was erroneous and that more caution is needed in the interpretation of classifier error rate in this context.

The information theoretic approach based on minimization of stochastic complexity adopted here [24,39–41] is closely related to the fully Bayesian approach, where a comparable model would lead to a posterior distribution over the possible combinations of classifications and expansion structures. Our current approach generalizes previous “naive classifiers” using class-conditional probability distributions expressing independence between features in [31,42], which are also trained by minimization of stochastic complexity. Similarly to the Bayesian modeling paradigm, SC enforces a trade-off between descriptonal/predictive accuracy and modeling complexity.

Our result on stochastic complexity for class-conditional probability distributions factorized along a (rooted) tree, whose nodes correspond to the components of a binary vector, was obtained by applying the results in [23,24] to a Chow expansion of the joint probability integrated with respect to Jeffreys’ prior. Generally, minimization of stochastic complexity corresponds in many cases to the minimum description length (MDL) principle of model choice. MDL principle is discussed for learning of the structure of graphs in [43,44], while surveys and tutorials of algorithms and techniques for learning graph structure from data are given in [45,46].

The procedure of learning trees from data was first presented in [12]. In this procedure the mutual information between all pairs of nodes is computed using the relevant sample frequencies and the best tree is selected as the one that gives the maximum overall mutual information. This is in fact a maximum likelihood estimate of the tree, the asymptotic consistency of which was proved in [47] for increasing sets of independent samples from a tree dependent distribution. The procedure was extended by Suzuki [28], who observed the connection of the Chow-Liu estimate to MDL. The techniques in [44] do not lead to this, as pointed out by Suzuki [28]. The procedure of structure learning to be applied here is, as

far as the probability distributions involved are concerned, closely related to the Bayesian algorithms in [20,21,48]. Learning of graphs from data using the search algorithms of Cooper and Herskovits is an NP-complete problem [49].

The deterministic algorithms for learning the population structure minimizing stochastic complexity that were introduced the previous section can be considered as relatively implementation-friendly, although they are still considerably computation intensive as the number of samples and marker loci increase. Given that such algorithms typically only converge to local minima when the model structure and topology of the search space are complex, it would be necessary to execute the algorithms multiple times from different random starting configurations to gain information about the stability of the learned optimal structures. Furthermore, since the stochastic complexities for any two model structures can be analytically compared, the difference in the optimal data encoding efficiency can be easily compared over multiple runs of the algorithms.

An alternative to the deterministic learning algorithms considered in this work would be to consider a family of Monte Carlo algorithms to either approximate the SC optimal population structure or to perform a fully Bayesian analysis where the posterior distribution over the population structures is approximated. We have earlier considered Markov chain Monte Carlo (MCMC)-based learning of unsupervised classification and graphical models [50–52], and in particular demonstrated that standard reversible Metropolis-Hastings algorithms may dramatically fail when the level of complexity of the considered models is very high. To resolve this issue, Corander *et al.* [50] introduced a parallel non-reversible MCMC algorithm for Bayesian model learning where the topology of the model space in combination with the probabilistic search operators is not allowed to influence the acceptance ratio of a Metropolis-Hastings algorithm. This strategy was illustrated to be much more fruitful than a standard reversible MCMC algorithm for learning a large dimensional unsupervised classification model. A particular strength of the non-reversible algorithm is that it enables more freedom in the design of the search operators utilized in the proposal mechanism, since the proposal probabilities need not be calculated explicitly. On the other hand, the currently considered learning problem is so complex in general, that any realistic implementation of a stochastic learning algorithm must be done within a true parallel computing environment to prevent the computation times becoming prohibitive in practice. Our future aim is to implement such algorithms and the deterministic algorithms considered in this work to compare their relative levels of performance for solving the learning task of unsupervised classification augmented with trees. Also, an interesting generalization of the introduced linkage modeling framework would be to consider multi-allelic loci as well as data from diploid and tetraploid organisms.

Acknowledgements

Work of JC was supported by ERC project no. 239784. The authors would like to thank two anonymous reviewers and Yaqiong Cui for comments that enabled us to improve the original version of the article.

References

1. Ewens, W.J. *Mathematical Population Genetics*, 2nd ed.; Springer-Verlag: New York, NY, USA, 2004.
2. Nagylaki, T. *Theoretical Population Genetics*; Springer-Verlag: Berlin, Germany, 1992.
3. Pritchard, J.K.; Stephens, M.; Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics* **2000**, *155*, 945-959.
4. Dawson, K.J.; Belkhir, K. A Bayesian approach to the identification of panmictic populations and the assignment of individuals. *Genet. Res.* **2001**, *78*, 59-77.
5. Corander, J.; Waldmann, P.; Sillanpää, M.J. Bayesian analysis of genetic differentiation between populations. *Genetics* **2003**, *163*, 367-374.
6. Corander J.; Marttinen, P. Bayesian identification of admixture events using multi-locus molecular markers. *Mol. Ecol.* **2006**, *15*, 2833-2843.
7. Corander, J.; Gyllenberg, M.; Koski, T. Random Partition models and Exchangeability for Bayesian Identification of Population Structure. *Bull. Math. Biol.* **2007**, *69*, 797-815.
8. Falush, D.; Stephens, M.; Pritchard, J.K. Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. *Genetics* **2003**, *164*, 1567-1587.
9. Guillot, G.; Estoup, A.; Mortier, F.; Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics* **2005**, *170*, 1261-1280.
10. Guillot, G.; Leblois, R.; Coulon, A.; Frantz, A.C. Statistical methods in spatial genetics. *Mol. Ecol.* **2010**, *18*, 4734-4756.
11. Gyllenberg, M.; Carlsson, J.; Koski, T. Bayesian Network Classification of Binarized DNA Fingerprinting Patterns. In *Mathematical Modelling and Computing in Biology and Medicine*; Capasso, V. Ed.; Progetto Leonardo: Bologna, Italy, 2003, pp. 60-66.
12. Chow, C.K.; Liu, C.N. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. Inf. Theory* **1968**, *14*, 462-467.
13. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian Network Classifiers. *Mach. Learn.* **1997**, *29*, 1-36.
14. Corander J.; Tang, J. Bayesian analysis of population structure based on linked molecular information. *Math. Biosci.* **2007**, *205*, 19-31.
15. Cowell, R.G.; Dawid, A.P.; Lauritzen, S.L.; Spiegelhalter, D.J. *Probabilistic Networks and Expert Systems*; Springer-Verlag: New York, NY, USA, 1999.
16. Koski, T.; Noble, J.N. *Bayesian Networks: an Introduction*; Wiley: Chichester, UK, 2009.
17. Meil, M.; Jordan, M.I. Learning with Mixtures of Trees. *J. Mach. Learn. Res.* **2000**, *1*, 1-48.
18. Pearl, J. *Probabilistic Reasoning in Intelligent Systems*; Morgan Kaufmann: San Francisco, CA, USA, 1988.
19. Becker, A.; Geiger, D.; Meek, C. Perfect Tree-like Markovian Distributions. *Proc. 16th Conf. Uncertainty in Artificial Intelligence* **2000**, 19-23.
20. Heckerman, D.; Geiger, D.; Chickering, D.M. Learning Bayesian Networks: The combination of knowledge and statistical data. *Mach. Learn.* **1995**, *20*, 197-243.

21. Heckerman, D.; Geiger, D.; Chickering, D.M. Likelihoods and Parameter Priors for Bayesian Networks. *Microsoft Res. Tech. Rep.* MSR-TR-95-54.
22. Duda, R.O.; Hart, P.E. *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973.
23. Clarke, B.S.; Barron, A.R. Jeffreys' prior is asymptotically least favorable under entropy risk. *J. Stat. Planning Inference* **1994**, *41*, 37-60.
24. Rissanen, J. Fisher Information and Stochastic Complexity. *IEEE Trans. Inf. Theory* **1996**, *42*, 40-47.
25. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley: New York, NY, USA, 1991.
26. Gyllenberg, M.; Koski, T. Bayesian Predictiveness, Exchangeability and Sufficiency in Bacterial Taxonomy. *Math. Biosci.* **2002**, *177 & 178*, 161-184.
27. DeGroot, M.H. *Optimal Statistical Decisions*. McGraw-Hill: New York, NY, USA, 1970.
28. Suzuki, J. Learning Bayesian Belief Networks Based on the Minimum Description Length Principle: Basic Properties. *IEICE Trans. Fundamentals* **1999**, *82*, 2237-2245.
29. Kučera, L. *Combinatorial Algorithms*; Adam Hilger: Bristol, UK, 1990.
30. Schwartz, G. Estimating the Dimension of a Model. *Ann. Statist.* **1978**, *6*, 461-464.
31. Gyllenberg, M.; Koski, T.; Verlaan, M. Classification of Binary Vectors by Stochastic Complexity. *J. Multiv. Analysis* **1997**, *63*, 47-72.
32. Kass, R.E.; Wasserman, L. A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwartz criterion. *J. Amer. Stat. Assoc.* **1995**, *90*, 928-934.
33. Drton, M.; Sturmfels B.; Sullivant, S. *Lectures on Algebraic Statistics*; Birkhäuser: Basel, Switzerland, 2005.
34. Rusakov, D.; Geiger, D. Asymptotic Model Selection for Naive Bayesian Networks. *J. Mach. Learn. Res.* **2005**, *6*, 1-35.
35. Biernacki, C.; Celeux, G.; Covaert, G. Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood. *IEEE Trans. Patt. Anal. Mach. Intel.* **2000**, *28*, 719-725.
36. Haughton, D.M.A. On the Choice of the Model to Fit Data from an Exponential Family. *Ann. Statist.* **1988**, *16*, 342-355.
37. Wong, S.K.M.; Poon, F.C.S. Comments on the Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. Patt. Anal. Mach. Intel.* **1989**, *11*, 333-335.
38. Balagani, K.S.; Phoha, V.V. On the Relationship between Dependence Tree Classification Error and Bayes Error Rate. *IEEE Trans. Patt. Anal. Mach. Intel.* **2007**, *29*, 1866-1868.
39. Rissanen, J. Stochastic Complexity in Learning. *J. Comp. System Sci.* **1997**, *55*, 89-95.
40. Vitányi, P.M.B.; Li, M. Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity. *IEEE Trans. Inf. Theory* **2000**, *46*, 446-464.
41. Yamanishi, J.K. A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Trans. Inf. Theory* **1998**, *44*, 1424-1439.
42. Gyllenberg, M.; Koski, T.; Lund, T.; Gyllenberg, H.G. Bayesian Predictive Identification and Cumulative Classification of Bacteria. *Bull. Math. Biol.* **1999**, *61*, 85-111.
43. Friedman, N.; Goldszmidt, M. Learning Bayesian Networks with Local Structure. In *Learning in Graphical Models*; Jordan, M. Ed.; MIT Press: Cambridge, MA, USA, 1997; pp. 421-459.

44. Lam, W.; Bacchus, F. Learning Bayesian Belief Networks: An Approach Based on the MDL Principle. *Comput. Intel.* **1994**, *10*, 269-293.
45. Buntine, W. A guide to the literature on learning probabilistic networks from data. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 195-210.
46. Sangüesa, R.; Cortés, U. Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks. *AI Commun.* **1997**, *10*, 31-61.
47. Chow, C.K.; Wagner, T.J. Consistency of an estimate of tree-dependent probability distributions. *IEEE Trans. Inf. Theory* **1973**, *19*, 369-371.
48. Cooper, G.F.; Herskovits, E. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Mach. Learn.* **1992**, *9*, 309-347.
49. Chickering, D.M. Learning Bayesian Networks is NP-Complete. In *Learning from Data. Artificial Intelligence and Statistics*; V. Fisher, D.; Lenz, H-J. Eds.; Springer-Verlag: New York, NY, USA, 1996; pp. 121-130.
50. Corander, J.; Gyllenberg, M.; Koski, T. Bayesian model learning based on a parallel MCMC strategy. *Stat. Comput.* **2006**, *16*, 355-362.
51. Corander, J.; Ekdahl, M.; Koski, T. Parallel interacting MCMC for learning of topologies of graphical models. *Data Mining Knowl. Discovery* **2008**, *17*, 431-456.
52. Corander, J.; Gyllenberg, M.; Koski, T. Bayesian unsupervised classification framework based on stochastic partitions of data and a parallel search strategy. *Adv. Data Anal. Classification* **2009**, *3*, 3-24.

Appendices

A.1. Prior predictive data distributions under Chow expansion

The integral on the right hand side of (9) will by force of (10) be factorized as

$$P(X^t|\Pi) = I_1 \cdot I_2 \cdot I_3, \quad (39)$$

where

$$I_1 = \int_0^1 \theta_1^{n_1} (1 - \theta_1)^{t-n_1} h(\theta_1) d\theta_1, \quad (40)$$

$$I_2 = \prod_{i=2}^d \int_0^1 \theta_i^{n_i(1,1)} (1 - \theta_i)^{n_i(0,1)} h(\theta_i) d\theta_i, \quad (41)$$

and

$$I_3 = \prod_{i=2}^d \int_0^1 \phi_i^{n_i(1,0)} (1 - \phi_i)^{n_i(0,0)} z(\phi_i) d\phi_i. \quad (42)$$

There is an explicit expression for each of the factors I_1 , I_2 and I_3 , if the prior densities $h(\cdot)$ and $z(\cdot)$ are Beta densities, e.g.,

$$h(\theta) = \begin{cases} \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \theta^{\alpha_1-1} (1 - \theta)^{\alpha_2-1} & 0 \leq \theta \leq 1 \\ 0 & \text{elsewhere,} \end{cases} \quad (43)$$

where $\alpha_i > 0$. Then, θ has a $\text{Be}(\alpha_1, \alpha_2)$ distribution. Using the Beta integral

$$\int_0^1 \theta^{\alpha_1-1} (1-\theta)^{\alpha_2-1} d\theta = \frac{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)}$$

we obtain, e.g., in each factor of I_2 in (41)

$$\begin{aligned} & \int_0^1 \theta_i^{n_i(1,1)} (1-\theta_i)^{n_i(0,1)} h(\theta_i) d\theta_i = \\ & \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \frac{\Gamma(n_i(1,1) + \alpha_1) \cdot \Gamma(n_i(0,1) + \alpha_2)}{\Gamma(n_i(1,1) + n_i(0,1) + \alpha_1 + \alpha_2)}. \end{aligned}$$

Thus we have

$$\begin{aligned} I_2 &= \prod_{i=2}^d \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \frac{\Gamma(n_i(1,1) + \alpha_1) \cdot \Gamma(n_i(0,1) + \alpha_2)}{\Gamma(n_i(1,1) + n_i(0,1) + \alpha_1 + \alpha_2)}, \\ I_3 &= \prod_{i=2}^d \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \frac{\Gamma(n_i(1,0) + \alpha_1) \cdot \Gamma(n_i(0,0) + \alpha_2)}{\Gamma(n_i(1,0) + n_i(0,0) + \alpha_1 + \alpha_2)}, \end{aligned}$$

as well as

$$I_1 = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \frac{\Gamma(n_1 + \alpha_1) \cdot \Gamma(t - n_1 + \alpha_2)}{\Gamma(t + \alpha_1 + \alpha_2)}.$$

A.2. Asymptotic expansion of the stochastic complexity for a Chow expansion

Assuming $\alpha_1 = \alpha_2 = 1/2$ we obtain the generic term (denoted by $E_{1/2}^{(2)}$) in I_2 (see (11)) as

$$\begin{aligned} E_{1/2}^{(2)} &\equiv \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \frac{\Gamma(n_i(1,1) + \alpha_1) \cdot \Gamma(n_i(0,1) + \alpha_2)}{\Gamma(n_i(1,1) + n_i(0,1) + \alpha_1 + \alpha_2)} \\ &= \frac{1}{\pi} \frac{\Gamma(n_i(1,1) + 1/2) \cdot \Gamma(n_i(0,1) + 1/2)}{\Gamma(n_i(1,1) + n_i(0,1) + 1)}. \end{aligned}$$

By invoking Stirling's formula in a straightforward calculation for $-\log E_{1/2}^{(2)}$ in $-\log I_2$ in (11), this entails

$$-\log E_{1/2}^{(2)} = (n_i(1,1) + n_i(0,1)) h(\hat{\theta}_i^{(1)}) + \frac{1}{2} \log(n_i(1,1) + n_i(0,1)) + C, \quad (44)$$

where C is a bounded in t and $h(x) = -x \log x - (1-x) \log(1-x)$, $0 \leq x \leq 1$, the binary entropy function (in natural logarithms) of the empirical distribution

$$(\hat{\theta}_i^{(1)}, 1 - \hat{\theta}_i^{(1)}) = \left(\frac{n_i(1,1)}{n_i(1,1) + n_i(0,1)}, \frac{n_i(0,1)}{n_i(1,1) + n_i(0,1)} \right). \quad (45)$$

Here $\hat{\theta}_i^{(1)}$ is the maximum likelihood estimate (based on X^t) of $\theta_i^{(1)} = P(x_i = 1 | x_{\Pi(i)} = 1)$.

For a generic term (denoted by $-\log E_{1/2}^{(3)}$) in $-\log I_3$ in (11) we obtain in the same way

$$-\log E_{1/2}^{(3)} = (n_i(1,0) + n_i(0,0)) h(\hat{\theta}_i^{(0)}) + \frac{1}{2} \log(n_i(1,0) + n_i(0,0)) + C, \quad (46)$$

where $\hat{\theta}_i^{(0)}$ is the maximum likelihood estimate of $\theta_i^{(0)} = P(x_i = 1 | x_{\Pi(i)} = 0)$.

Next we consider the result of inserting the terms

$$(n_i(1,1) + n_i(0,1)) h(\hat{\theta}_i^{(1)})$$

and

$$(n_i(1, 0) + n_i(0, 0)) h(\hat{\theta}_i^{(0)})$$

in the right hand side of (11). This gives the following expression

$$\sum_{i=2}^d \left[(n_i(1, 1) + n_i(0, 1)) h(\hat{\theta}_i^{(1)}) + (n_i(1, 0) + n_i(0, 0)) h(\hat{\theta}_i^{(0)}) \right].$$

The generic term in the sum is

$$(n_i(1, 1) + n_i(0, 1)) h(\hat{\theta}_i^{(1)}) + (n_i(1, 0) + n_i(0, 0)) h(\hat{\theta}_i^{(0)}).$$

This expression is by definition of the binary entropy function $h(x)$ equal to

$$\begin{aligned} &= -n_i(1, 1) \log \left(\frac{n_i(1, 1)}{n_i(1, 1) + n_i(0, 1)} \right) - n_i(0, 1) \log \left(\frac{n_i(0, 1)}{n_i(1, 1) + n_i(0, 1)} \right) \\ &- n_i(1, 0) \log \left(\frac{n_i(1, 0)}{n_i(1, 0) + n_i(0, 0)} \right) - n_i(0, 0) \log \left(\frac{n_i(0, 0)}{n_i(1, 0) + n_i(0, 0)} \right). \end{aligned} \quad (47)$$

Let us introduce the auxiliary quantities

$$n_{\Pi(i)}(1) = n_i(1, 1) + n_i(0, 1), n_{\Pi(i)}(0) = n_i(1, 0) + n_i(0, 0), \quad (48)$$

and

$$n_i(1) = n_i(1, 1) + n_i(1, 0), n_i(0) = n_i(0, 1) + n_i(0, 0). \quad (49)$$

Then we have as an identity from the right hand side of (47)

$$\begin{aligned} &-n_i(1, 1) \log \left(\frac{n_i(1, 1)}{n_i(1, 1) + n_i(0, 1)} \right) - n_i(0, 1) \log \left(\frac{n_i(0, 1)}{n_i(1, 1) + n_i(0, 1)} \right) \\ &-n_i(1, 0) \log \left(\frac{n_i(1, 0)}{n_i(1, 0) + n_i(0, 0)} \right) - n_i(0, 0) \log \left(\frac{n_i(0, 0)}{n_i(1, 0) + n_i(0, 0)} \right) = \\ &= -n_i(1, 1) \log \left(\frac{n_i(1, 1)/t \cdot n_i(1)/t}{n_{\Pi(i)}(1)/t \cdot n_i(1)/t} \right) - n_i(0, 1) \log \left(\frac{n_i(0, 1)/t \cdot n_i(0)/t}{n_{\Pi(i)}(1)/t \cdot n_i(0)/t} \right) \\ &-n_i(1, 0) \log \left(\frac{n_i(1, 0)/t \cdot n_i(1)/t}{n_{\Pi(i)}(0)/t \cdot n_i(1)/t} \right) - n_i(0, 0) \log \left(\frac{n_i(0, 0)/t \cdot n_i(0)/t}{n_{\Pi(i)}(0)/t \cdot n_i(0)/t} \right). \end{aligned}$$

The quantities in the right hand side of the last equality can be regrouped as

$$\begin{aligned} &-n_i(1, 1) \log (n_i(1)/t) - n_i(0, 1) \log (n_i(0)/t) \\ &-n_i(1, 0) \log (n_i(1)/t) - n_i(0, 0) \log (n_i(0)/t) \\ &-n_i(1, 1) \log \left(\frac{n_i(1, 1)/t}{n_{\Pi(i)}(1)/t \cdot n_i(1)/t} \right) - n_i(0, 1) \log \left(\frac{n_i(0, 1)/t}{n_{\Pi(i)}(1)/t \cdot n_i(0)/t} \right) \\ &-n_i(1, 0) \log \left(\frac{n_i(1, 0)/t}{n_{\Pi(i)}(0)/t \cdot n_i(1)/t} \right) - n_i(0, 0) \log \left(\frac{n_i(0, 0)/t}{n_{\Pi(i)}(0)/t \cdot n_i(0)/t} \right). \end{aligned}$$

The first four terms are equal to

$$\begin{aligned} & -n_i(1, 1) \log (n_i(1)/t) - n_i(0, 1) \log (n_i(0)/t) \\ & -n_i(1, 0) \log (n_i(1)/t) - n_i(0, 0) \log (n_i(0)/t) = \\ & -(n_i(1, 1) + n_i(1, 0)) \log (n_i(1)/t) - (n_i(0, 1) + n_i(0, 0)) \log (n_i(0)/t) = \\ & -n_i(1) \log (n_i(1)/t) - n_i(0) \log (n_i(0)/t) = t \cdot h(n_i(1)/t). \end{aligned}$$

The remaining terms are

$$\begin{aligned} & -n_i(1, 1) \log \left(\frac{n_i(1, 1)/t}{n_{\Pi(i)}(1)/t \cdot n_i(1)/t} \right) - n_i(0, 1) \log \left(\frac{n_i(0, 1)/t}{n_{\Pi(i)}(1)/t \cdot n_i(0)/t} \right) \\ & -n_i(1, 0) \log \left(\frac{n_i(1, 0)/t}{n_{\Pi(i)}(0)/t \cdot n_i(1)/t} \right) - n_i(0, 0) \log \left(\frac{n_i(0, 0)/t}{n_{\Pi(i)}(0)/t \cdot n_i(0)/t} \right) = \\ & -t \cdot I_{i, \Pi(i)}, \end{aligned}$$

where we have defined the *mutual information* [25] between X_i and $X_{\Pi(i)}$ by

$$I_{i, \Pi(i)} = \sum_{u=0}^1 \sum_{v=0}^1 \hat{P}_{i, \Pi(i)}(u, v) \log \frac{\hat{P}_{i, \Pi(i)}(u, v)}{\hat{P}_i(u) \cdot \hat{P}_{\Pi(i)}(v)} \quad (50)$$

using the maximum likelihood estimates (*i.e.* observed relative frequencies) of the two dimensional distributions $P_{i, \Pi(i)}(u, v)$ as well as of the marginal distributions $P_i(u)$ and $P_{\Pi(i)}(v)$.

© 2010 by the authors; licensee MDPI Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license <http://creativecommons.org/licenses/by/3.0/>.