Using Colored Petri Nets to Construct Coalescent Hidden Markov Models: Automatic Translation from Demographic Specifications to Efficient Inference Methods

Thomas Mailund[1], Anders E. Halager[1,2], and Michael Westergaard[3]

[1] Bioinformatics Research Center, Aarhus University, Denmark
[2] Department of Computer Science, Aarhus University, Denmark
[3] Department of Mathematics and Computer Science,
Eindhoven University of Technology, The Netherlands

Abstract. Biotechnological improvements over the last decade has made it economically and technologically feasible to collect large DNA sequence data from many closely related species. This enables us to study the detailed evolutionary history of recent speciation and demographics. Sophisticated statistical methods are needed, however, to extract the information that DNA sequences hold, and a limiting factor in this is dealing with the large state space that the ancestry of large DNA sequences spans. Recently a new analysis method, CoalHMMs, has been developed, that makes it computationally feasible to scan full genome sequences – the complete genetic information of a species – and extract genetic histories from this. Applying this methodology, however, requires that the full state space of ancestral histories can be constructed. This is not feasible to do manually, but by applying formal methods such as Petri nets it is possible to build sophisticated evolutionary histories and automatically derive the analysis models needed. In this paper we describe how to use colored stochastic Petri nets to build CoalHMMs for complex demographic scenarios.

1 Introduction

Biotechnological advances over the last decade have dramatically reduced the cost of obtaining the full genetic material of an individual – the full genome – and genomes from many closely related species are now available. For example, one or more genomes have been sequenced for each species of the great apes, the closest related species to humans. This puts us in the unique position to learn much more about human evolutionary history over the last 15 million years than what has previously been gleaned from fossils and from single gene studies.

Computational approaches to studying biology enables sophisticated analysis and provide the only feasible approach to analysis for very large data sets, such as full genome sequences. Whole genome comparisons hold the key to decipher the speciation process, selection and demographic changes in human and great ape

S. Haddad and L. Pomello (Eds.): PETRI NETS 2012, LNCS 7347, pp. 32–50, 2012.

history, but analysis methods that are statistical powerful and computationally efficient are still in their infancy.

Sequential Markov Coalescent (SMC) and its inference method Coalescent hidden Markov models (CoalHMMs) [5,8,9,12,18,22,24] is a recently developed methodology for analyzing genome relationships and make inference of speciation divergence and the mechanisms involved in speciation. CoalHMMs combine the so-called "coalescence process" model of population genetics [11] with the computational efficient statistical tool "hidden Markov models" [7] and provides the first approach to analyze the speciation process computationally scalable to whole-genome analysis. CoalHMMs model the dependence of the genealogies (tree relationships) between neighboring nucleotides along a genomic sequence as a function of the events of coalescence and recombination in the history of the sequences, and can analyze samples of entire genomes appropriately aligned.

The first CoalHMMs were designed to estimate split times and genetic diversity in the species ancestral to human, chimpanzee and gorilla by analyzing patterns of incomplete lineage sorting – i.e. patterns of genealogies inconsistent with the species phylogeny caused by deep coalescences [8,12]. The same models were later used to analyze the complete orangutan genome [17] and gave insight into the evolutionary forces forming the great ape genomes [13]. The models have also been applied to the gorilla genome [29] and bonobo genome [25] further illuminating the evolution of our own species by comparing our genome with our closest ape relatives. A different approach to CoalHMMs was recently used to infer demographic parameters of the human species [16].

A major limitation of the initial CoalHMM methods is that they do not generalize to comprise complex demographic and speciation scenarios. The methods strongly depend on patterns of incomplete lineage sorting and do not allow for complex population structures, population size bottlenecks, gene flow etc. This was amended in the method used for analyzing the orangutan subspecies [18]. Here, a mathematical model based on continuous time Markov chains (CTMCs) was used to explicitly model the probability of changes in genealogies along a genome sequence, using exact calculations from the coalescence process. While this first CTMC based method is rather simple, only capturing changes in coalescence time between two genomes, the strengths of the CTMC approach is that, in theory, it generalizes to a large variety of scenarios.

Constructing CTMC models of complex demographic scenarios, however, is at best tedious and error prone, considering the large state space of these models, and it is unlikely that they can be constructed correctly by hand. In this paper we propose using colored Petri nets (CPN) [15] as a formal method for specifying genetic models and give algorithms for translating the state space of such models into CoalHMMs.

In the next section, we will provide background information for the coalescent process and present a CPN model of the coalescence process. In the following section we describe how the coalescent CPN model can be used to define a Markov model along a genome, approximating the coalescence process. This is similar to the construction of a Markov chain from a stochastic Petri net [21].

We then present results from our prototype implementation, and finally draw our conclusions.

2 Modeling the Coalescent Process

The general idea behind coalescent hidden Markov models is to approximate the coalescent process by a Markov model along a genomic alignment. Below we first present the coalescence process and then present a CPN model of the coalescence process over two neighboring nucleotides.

2.1 The Coalescent Process

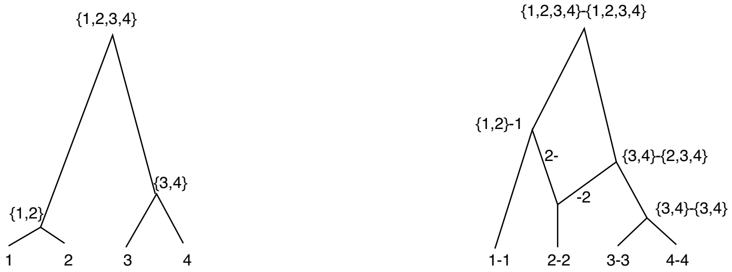
The coalescent process [11] is a statistical model describing the genetic relationship of a sample of genes. The coalescent process assumes that k genes have been sampled in a population, and models how their ancestry (or "genealogy") could be, providing probabilities to different scenarios of the genes ancestry from which a number of properties of the population can be inferred.

The process runs backwards in time, and in its simplest form each pair of genes can coalesce with a fixed rate. When two genes coalesce, it models the time where they last shared an ancestor (known as the most recent common ancestor, or MRCA, of the two genes). After a pair of genes have coalesced, they are replaced by a gene representing their MRCA, and the process continues further back in time, now with $k - 1$ genes. The process is continued until all genes have coalesced, i.e. when $k = 1$. A run of the coalescent process corresponds to a tree, where the order in which different genes coalesce determines the topology and the time in which genes coalesce determines the branch lengths, see Fig. 1 (a).

By treating the process as a continuous time Markov chain, each tree can be assigned a probability, and by placing mutations on the tree we can compute the probability that a given tree gave rise to the observed genes. From this we can get the joint probability of the tree and the observed genes, and use this to make statistical inference. Since the true ancestry of genes is unknown, and in general unknowable, statistical inference based on the coalescence process involves integrating over all trees, either explicitly (for small k) or with statistical Monte Carlo integration (for larger k).

A simple tree relationship for genes, however, is an inaccurate model of species with two genders. Sex cells in species with two genders are constructed as "recombinations" of the genetic material inherited by each parent. In the coalescent process, this is modeled by adding a second type of event. Each gene can undergo recombination, in which case the gene is split in two at a random point, the left and right side of the recombination point. The process is then continues with $k + 1$ genes as the left and right part of the recombined gene is assumed to have independent ancestries.

A run of the coalescent process with recombination can no longer be represented as a tree but instead a directed acyclic graph, known as the *ancestral recombination graph* or ARG, see Fig. 1 (b). Scanning from left to right along



(a) A coalescence tree over a sample of size 4

(b) An ancestral recombination graph over a sample of size 4 of sequences of length 2

Fig. 1. A coalescence tree and an ancestral recombination graph. (a) A coalescence tree, where first genes 1 and 2 coalesce into their most recent common ancestor, $\{1, 2\}$, then genes 3 and 4 coalesce into their most recent common ancestor $\{3, 4\}$ and finally all genes coalesce into the grand most recent common ancestor. (b) An ancestral recombination graph of four sequences of length two. First genes 3 and 4 coalesce, where both their left and right nucleotide find an ancestor at the same time. Then gene 2 recombines, leading to independent genes for its left and right nucleotide. The right nucleotide of gene 2 coalesce with the ancestor of genes 3 and 4 while the left nucleotide of gene 2 coalesce with gene 1, before all genes find their most recent common ancestor. The left and right nucleotide in this example have different genealogies, with the left having topology $((1, 2), (3, 4))$ and the right having topology $(1, (2, (3, 4)))$.

the sampled genes, at each point the ancestry of the genes will be a tree, but the trees can change whenever a recombination point is seen. The tree at each point is known as a local genealogy while the ARG is known as the global genealogy of the genes.

The state space of possible ARGs for a gene sample is generally intractable for all but the smallest samples [30] even for statistical integration, and to deal with large sample sizes or long gene sequences approximations to the process is necessary. One such approximation is assuming that the relationship between genealogies is Markov along the genes [1, 20], an assumption that greatly reduces the complexity of the process. Assuming the Markov property essentially means that we only need to model pairs of nucleotides rather than the full DNA sequence, since the probability of a sequence can be specified through all the pairwise probabilities.

2.2 A Colored Petri Net Model for Pairwise Genealogies

While the coalescence process is difficult to make inference from, the rules for how the process generates genealogies are straightforward and can be expressed as a very simple colored Petri net. The way the coalescence process treats genes as independent items with events that can affect one or two genes maps straightforwardly to a CPN model where genes become tokens and coalescence and

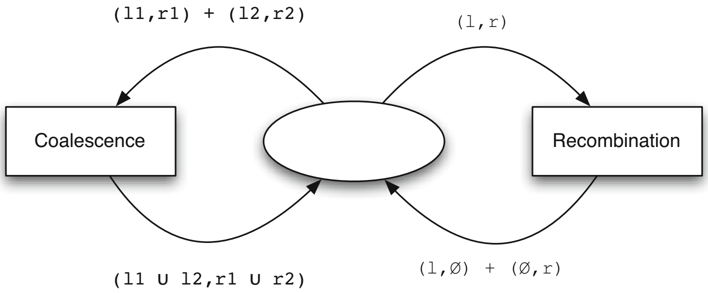


Fig. 2. CPN model of the basic two-nucleotide coalescence. This rather simple colored Petri net can construct all two-nucleotide coalescence runs for any number of samples in a single population. The set of genes in the process are represented as tokens on the single place, where each token contains a pair of sets of sampled genes. The pair represent the left and right nucleotide in the gene, and the sets the genes or most recent common ancestor of a set of genes. A coalescence event combines the left and right sets of the genes, while a recombination breaks up one gene into two: the left and right nucleotide of the original gene.

recombination events become transitions. Such a CPN model is shown in Fig. 2. The CPN model has a single place, containing the genes of the process, and two transitions modeling the two operations Coalescence and Recombination. The tokens on the single place consists of pairs – the left and right nucleotide of the genes – and each nucleotide will contain the set of original sampled genes. The initial marking consists of pairs $(\{i\},\{i\})$ for genes $i = 1,\ldots,k$

A run of this CPN, producing the ARG in Fig. 1 (b), would look like this:

State:	$1\lq(\{1\},\{1\}) + 1\lq(\{2\},\{2\}) + 1\lq(\{3\},\{3\}) + 1\lq(\{4\},\{4\})$
Binding:	$[\mathsf{Coalescence}; 1\lq(\{3\},\{3\}) + 1\lq(\{4\},\{4\})\rangle$
State:	$1\lq(\{1\},\{1\}) + 1\lq(\{2\},\{2\}) + 1\lq(\{3,4\},\{3,4\})$
Binding:	$[\mathsf{Recombination}; 1\lq(\{2\},\{2\})\rangle$
State:	$1\lq(\{1\},\{1\}) + 1\lq(\{2\},\emptyset) + 1\lq(\emptyset,\{2\}) + 1\lq(\{3,4\},\{3,4\})$
Binding	$[\mathsf{Coalescence}; 1\lq(\{3,4\},\{3,4\}) + 1\lq(\emptyset,\{2\})\rangle$
State	$1\lq(\{1\},\{1\}) + 1\lq(\{2\},\emptyset) + 1\lq(\{3,4\},\{2,3,4\})$
Binding:	$[\mathsf{Coalescence}; 1\lq(\{1\},\{1\}) + 1\lq(\{2\},\emptyset)\rangle$
State	$1\lq(\{1,2\},\{1\}) + 1\lq(\{3,4\},\{2,3,4\})$
Binding:	$[\mathsf{Coalescence}; 1\lq(\{1,2\},\{1\}) + 1\lq(\{3,4\},\{2,3,4\})\rangle$
State:	$1\lq(\{1,2,3,4\},\{1,2,3,4\})$

When we have different populations or different species, the probability of coalescing genes in different populations/species is zero, and we cannot model genes in this simple way. To model this, we can annotate tokens with populations and only allow Coalescence and Recombination to affect genes within a single population, but instead add a new event that migrates a gene from one population to another, see Fig. 3.

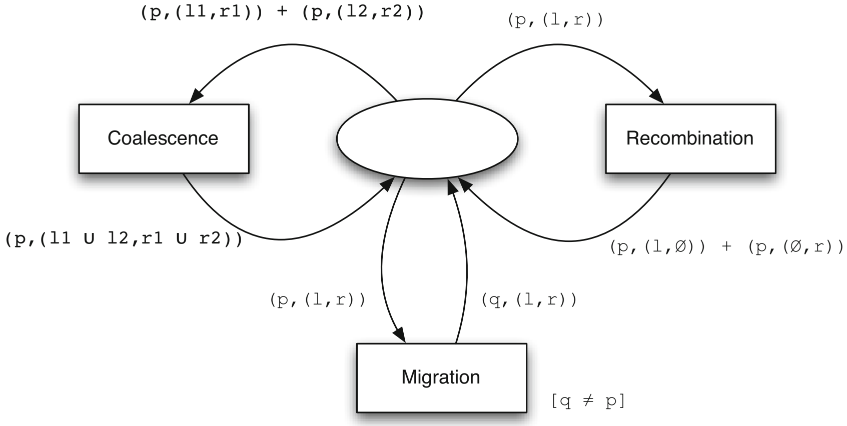


Fig. 3. CPN model with migration. To model different populations, we annotate each token with the population it belongs to. Coalescence events are only possible between lineages in the same population. Recombination, as well, although this only involves a single lineage so the difference is only seen in the arc annotation. To allow lineages to move from one population to another, a new transition is added that moves one lineage from one population to another.

2.3 Building Coalescent CTMCs from the Petri Net Specification

From the CPN specification we can build a state space capturing all possible ancestries of a sample. Our goal is to assign probabilities to all such ancestries. To do this, we consider the process a continuous time Markov chain (CTMC), and build the complete state space graph of the system. This corresponds to a matrix of rates between states where the rate between states is given by the type of transition in the CPN, and is similar to how a Markov chain is constructed from a Stochastic Petri net. We cannot use vanilla stochastic Petri nets, though as the transition rates depend also on the binding of the variables and we need to keep it symbolic for future estimation.

In terms of CTMC theory, what we construct is the instantaneous transition matrix, usually denoted Q and from this we can derive the probability of any run of the system. Obtaining a probabilistic model of the ancestries of a sample thus involves building the complete state space of the CPN model, translating this into a matrix of rates of transitions and considering this a CTMC rate matrix. For samples from a single population, we assign a fixed rate to transitions and recombinations (see Fig. 2), while for a scenario with multiple populations, we allow different coalescence rates for each population and different migration rates between different pairs of populations.

3 Constructing Sequential Markov Coalescent Models

The computational efficiency of CoalHMMs stems from assuming that the probability distribution of genealogies along a genome alignment is Markov: The probability of a local genealogy depends on its immediate neighbor, but not the more distant genealogies [1, 20, 22]. This way, the probability of a genealogy of the entire alignment can be specified from just the probability distribution of genealogies of two neighboring nucleotides [8, 18].

Let $\Pr(\mathcal{G}_L, \mathcal{G}_R)$ denote the joint probability of genealogies, $\mathcal{G}_L$ and $\mathcal{G}_R$ of two nucleotides L and R (left and right). If this probability can be efficiently computed, then the probability of a genealogy over L nucleotides, $\Pr(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_L)$ can efficiently be computed as $\Pr(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_L) = \Pr(\mathcal{G}_1) \prod_{i=1}^{L-1} \Pr(\mathcal{G}_{i+1} \mid \mathcal{G}_i)$ where $\Pr(\mathcal{G}_1) = \sum_g \Pr(\mathcal{G}_1, g)$ and $\Pr(\mathcal{G}_{i+1} \mid \mathcal{G}_i) = \Pr(\mathcal{G}_i, \mathcal{G}_{i+1}) / \Pr(\mathcal{G}_i)$.

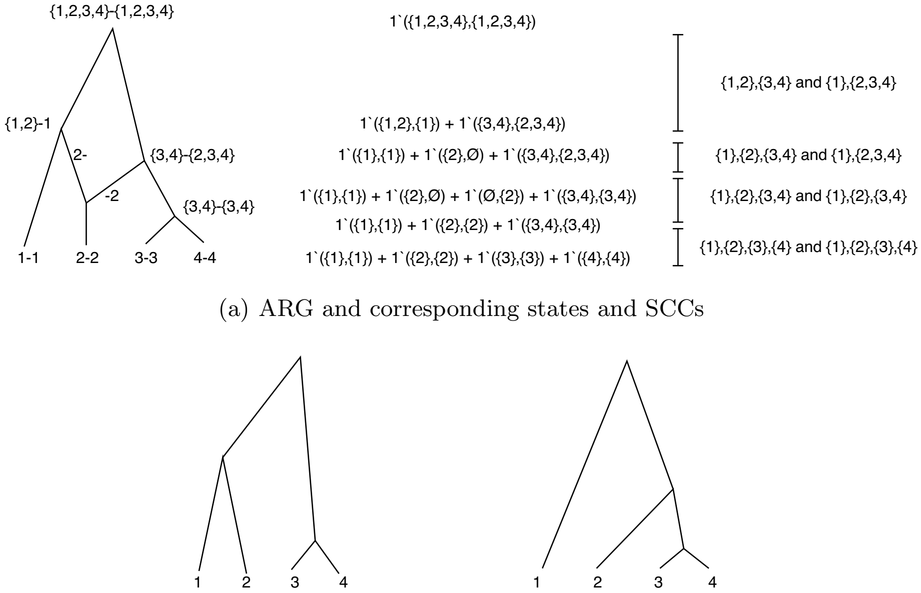
The key idea in Mailund *et al.* [18], that we generalize in this paper, was that these joint probabilities can be computed from a two-nucleotide CTMC. We can explicitly enumerate all possible states and state changes in the ancestry of two neighboring nucleotides, construct the corresponding CTMC, and obtain probabilities from this. Constructing the CTMC manually is feasible for small systems, as in Mailund *et al.* [18], but quickly becomes unmanageable. Below we show how the system can be constructed from a colored Petri net, and how the joint probability of a pair of genealogies can be algorithmically constructed from this.

To fully specify a CoalHMM we need to specify both transition and emission matrices (see Appendix A for the formal specification of a hidden Markov model), but since emission matrices can be computed using standard bioinformatics techniques, we will only focus on the transition matrix here. Constructing the transition matrix for the CoalHMM from the CTMC involves two steps: projecting a run of the CTMC onto the two neighboring genealogies, and discretizing time into time-intervals.

3.1 Projecting Runs of the CPN Model onto Pairs of Genealogies

A run of the CTMC involves coalescence events, recombination events and migration events. Of these, only coalescence events, where two lineages find a MRCA, are observable in the genealogies. All other events are important for computing the probability of the genealogies, but only the times of MRCAs are directly observable as genealogies and all other events should be integrated out when the probabilities of genealogies are computed.

The time points where two lineages find their MRCA corresponds to transitions in the CTMC state space where the system moves from one strongly connected component (SCC) to another since both migration and recombination events are reversible through a migration back or a coalescence of the two genes recombined, respectively. The genealogies of interest thus correspond to the paths in the SCC graph of the CTMC state space. Enumerating all paths in the SCC graph thus gives us all the genealogies to be considered, and there is a one-to-one correspondence between pairs of genealogies and paths in the SCC



(a) ARG and corresponding states and SCCs

(b) Left genealogy

(c) Right genealogy

Fig. 4. ARG, state space and strongly connected components of a run. (a) On the left the ARG from Fig. 1 (b). In the middle the CPN states corresponding to this ARG. On the right, the strongly connected components corresponding to this run of the CPN. The SCC is represented by the coalesced lineages on the left and right, respectively, and does not change due to the recombination event on the ARG. (b) The left genealogy of the ARG. (c) The right genealogy of the ARG.

graph. Fig. 4 shows a run of the CPN of a coalescent system. Here an ARG (from Fig. 1) is shown together with the states of the CPN that can produce this system, the SCC run of the system and the left and right genealogies of this run.

Paths in the SCC graph corresponds to pairs of genealogies and will be the state transitions in the hidden Markov model we construct. To exploit the efficient algorithms for HMMs we need to project the infinite state space of SCC paths onto a finite state space. We do this by discretizing time into a finite, fixed set of non-overlapping time intervals, $[\tau_0, \tau_1]$, $[\tau_1, \tau_2]$, up to $[\tau_{n-1}, \tau_n]$ with $\tau_0 = 0$ and $\tau_n = \infty$. We obtain finite state spaces by only considering which state the system is in at the time points between these intervals $(\tau_1, \tau_2, \ldots, \tau_{n-1})$.

We combine the discretized time with the valid SCC runs as follows. For any path through the SCC graph, $c_1, c_2, \ldots, c_n$, we assign time points to components $\tau_1 \leftrightarrow c_{i_1}, \tau_2 \leftrightarrow c_{i_2}, \ldots \tau_{n-1} \leftrightarrow c_{i_{n-1}}$ where $i_j \leq i_k$ for $j \leq k$. So, as an example, with three time intervals $[\tau_0 = 0, \tau_1]$, $[\tau_1, \tau_2]$ and $[\tau_2, \tau_3 = \infty]$ and an SCC path with two components, c_1, c_2, we would get the following three timed paths:

Interval	τ_1	Interval	τ_2	Interval
$[\tau_0, \tau_1]$	c_1	$[\tau_1, \tau_2]$	c_1	$[\tau_2, \tau_3]$
$[\tau_0, \tau_1]$	c_1	$[\tau_1, \tau_2]$	c_2	$[\tau_2, \tau_3]$
$[\tau_0, \tau_1]$	c_2	$[\tau_1, \tau_2]$	c_2	$[\tau_2, \tau_3]$

Notice that not all components need to be assigned a time point, and some can be assigned to several. This reflects that the system can move through several components within a single time interval and also stay in one component over several time intervals.

CTMC theory provides us with the mechanism for integrating over all paths leading from one state to another. If Q denotes the instantaneous rate matrix of the CTMC, then the probability of being in state s at time τ_i and state t at time τ_{i+1} is given by $P_{i,j}^{\tau_{i+1}-\tau_i}$ where $P^{\tau_{i+1}-\tau_i} = \exp\left(Q\left[\tau_{i+1} - \tau_i\right]\right)$ (where $\exp(M)$ denotes matrix exponentiation [23]). The probability of being in SCC c_i at time τ_i and SCC c_j at time τ_{i+1} is then computed by summing over all transitions from a state $s \in c_i$ to a state $t \in c_j$ in the time interval $[\tau_i, \tau_{i+1}]$: $\sum_{s\in c_i}\sum_{t\in c_j} P_{s,t}^{\tau_{i+1}-\tau_i}$. The probability of an entire SCC path assigned to time intervals is obtained by summing across all time intervals in this way (see Fig. 5), e.g. for the example above:

$$\Pr\left([\tau_0,\tau_1]\ c_1\ [\tau_1,\tau_2]\ c_1\ [\tau_2,\tau_3]\right) = \sum_{s\in c_1}\sum_{t\in c_1} P_{\iota,s}^{\tau_1}\cdot P_{s,t}^{\tau_2-\tau_1}$$

$$\Pr\left([\tau_0,\tau_1]\ c_1\ [\tau_1,\tau_2]\ c_2\ [\tau_2,\tau_3]\right) = \sum_{s\in c_1}\sum_{t\in c_2} P_{\iota,s}^{\tau_1}\cdot P_{s,t}^{\tau_2-\tau_1}$$

and

$$\Pr\left([\tau_0,\tau_1]\ c_2\ [\tau_1,\tau_2]\ c_2\ [\tau_2,\tau_3]\right) = \sum_{s\in c_2}\sum_{t\in c_2} P_{\iota,s}^{\tau_1}\cdot P_{s,t}^{\tau_2-\tau_1}$$

(notice changes in subscripts) where we assume that the system always starts in a fixed state ι.

For the general case $[\tau_0,\tau_1]\ c_{i_1}\ [\tau_1,\tau_2]\ c_{i_2}\ldots[\tau_{n-2},\tau_{n-1}]\ c_{i_{n-1}}\ [\tau_{n-1},\tau_n]$ this becomes

$$\sum_{s_1\in c_{i_1}}\sum_{s_2\in c_{i_2}}\cdots\sum_{s_{n-1}\in c_{i_{n-1}}} P_{\iota,s_1}^{\tau_1}\cdot P_{s_{i_1},s_{i_2}}^{\tau_2-\tau_1}\cdots P_{s_{n-2},s_{n-1}}^{\tau_{n-1}-\tau_{n-2}}$$

which is a sum of $|c_{i_1}|\times|c_{i_2}|\times\cdots\times|c_{i_{n-1}}|$ terms, where each term is a product of $n-1$ transition probabilities. To efficiently compute this for all paths, we rewrite this to

$$\sum_{s_1\in c_{i_1}} P_{\iota,s_1}^{\tau_1}\left(\sum_{s_2\in c_{i_2}} P_{s_{i_1},s_{i_2}}^{\tau_2-\tau_1}\left(\cdots\left(\sum_{s_{n-1}\in c_{i_{n-1}}} P_{s_{n-2},s_{n-1}}^{\tau_{n-1}-\tau_{n-2}}\right)\right)\cdots\right)$$

which we can compute inside-out for all paths using dynamic programming.

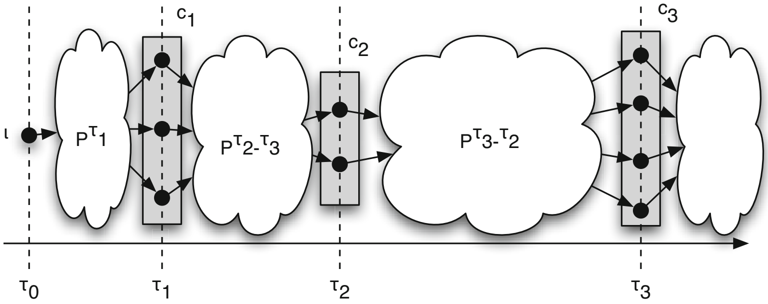


Fig. 5. Computing path probabilities. When computing the probability of the timed path $[\tau_0, \tau_1]$ c_1 $[\tau_1, \tau_2]$ c_2 $[\tau_2, \tau_3]$ c_3 $[\tau_3, \tau_4]$ we implicitly sum over all paths between the time interval breakpoints using CTMC transition probability matrices $P^{\tau_{i+1}-\tau_i}$ and explicitly sum over states in the strongly connected components at the breakpoints c_i.

3.2 Dealing with Different Demographic Epochs

When modeling the history of a set of genomes from different species, we need to consider different time period of their history. Consider for example a model of the ancestry of three different species, e.g. humans, chimpanzees and gorillas. At present, these are three different species that cannot exchange genes, but as we go back in time we first enter a period where humans and chimpanzees share an ancestral species, where they can exchange genes, and further back in time all three species share an ancestor where they exchange genes.

To deal with this we use what we called different "epochs". Each epoch corresponds to separate model in terms of transitions and transition rates, but all epochs for the same analysis can be embedded in the same (often large) state space, enabling us to map states between them. For the human, chimpanzee and gorilla example, we would have three populations/species and one sample from each. So the type used for lineages would have three colors (e.g. H, C and G for human, chimpanzee and gorilla) and the type used for populations also three colors. The space of all possible states would be all states that the CPN could be in. The different epochs would consist of restrictions to this state space, and typically we would never enumerate the full state space but only the sub-state spaces reachable in the different epochs.

A simple human, chimpanzee and gorilla model could have three epochs, one where all three species are isolated, one where humans and chimpanzees have found a common ancestor and one where all three African apes have found a common ancestor. This model will not allow migrations in any epochs. The first epoch will have each species in its own population, the second epoch would have humans and chimpanzees in the same population, and the third epoch would have all three species in the same population.

We construct the model by first constructing the state space of the first epoch, where the populations are H, C and G. We then take all reachable states in this system and maps H and C tokens to the same population, e.g. H, so tokens are mapped $(p,(l,r)) \mapsto (\mathsf{H},(l,r))$ whenever p is H or C. For the second epoch, we compute the state space of all states reachable from these mapped states (but not states from the first epoch where tokens can be in population C). For the third epoch we repeat this, but now mapping G populations to H as well.

When computing the probability of paths in the system, we add this projection of states as well. If the time point τ_i is between two different epochs, we use a matrix $P^{\tau_i-\tau_{i-1}} \cdot I_i$ instead of $P^{\tau_i-\tau_{i-1}}$ where I_i is a projection matrix mapping states from the epoch before τ_i to the epoch after τ_i. For the transition between the first and second epoch in the human, chimpanzee and gorilla example, this projection matrix would have a 1 in all entries where the states are equal exact for all C populations being set to H and 0 in all other entries, and for the projection from the second to the third epoch, the projection matrix would have a 1 in entries where the states are equal except that now G populations are set to H as well. The projection onto left and right genealogies, and the sums used for computing the probabilities of strongly connected components paths is not changed otherwise.

4 Results

The algorithm was implemented in the Python programming language and below we show results for state space construction, model fit and parameter estimation.

4.1 State Space Statistics

We constructed the state space and HMM transition matrix for a number of different configurations, varying the number of populations from one to three and varying the number of chromosomes from one to four. With one population there is a single time epoch, with two populations there are two epochs, one before and after the populations merge, and with three populations there are three time epochs: the first before any populations merge, the second after the first and second population merge, and the last when all three populations have merged.

Table 1 shows the size of the state spaces in the various configurations and epochs and the time it takes to construct the HMM transition matrix. The HMM construction time is split in three components: 1) the time it takes to construct the CTMC (i.e. build the state space of the CPN and translate it into a rate matrix), 2) pre-processing time for the HMM construct, involving building the SCC graph and assign all possible SCC paths to time intervals, and 3) the time it takes to construct the actual transition matrix, involving exponentiating rate matrices and summing over SCC paths. Of these three, the first two needs only be computed once for a given model, while the third needs to be recomputed whenever the parameters of the HMM changes, and must

Table 1. Summaries of the state space sizes, SCCs and construction time for both the state space and the hidden Markov model transition matrix. Configurations $n = i, j, k$ should be read as population one containing i chromosomes, population two containing j chromosomes and population three containing k chromosomes. Construction time is measured in seconds and – indicates that the computation was terminated before finishing.

	1st epoch		2nd epoch		3rd epoch		Construction time		
Configuration	States	SCCs	States	SCCs	States	SCCs	CTMC	Pre.	Trans.
1 population									
n = 1	2	1					0.00	0.00	0.01
n = 2	15	4					0.00	0.01	0.09
n = 3	203	25					0.03	3.44	18.02
n = 4	4 140	225					1.35	–	–
2 populations									
n = 1,1	4	1	15	4			0.00	0.02	0.08
n = 2,1	30	4	203	25			0.03	14.60	24.90
n = 3,1	406	25	203	25			1.71	–	–
n = 2,2	225	16	4 140	225			1.49	–	–
3 populations									
n = 1,1,1	8	1	30	4	203	25	0.11	19.75	21.87
n = 2,1,1	60	4	306	25	4 140	225	1.67	–	–

potentially be computed hundreds of times in a numerical optimization of the HMM likelihood.

The most time consuming part of constructing the HMM is clearly not constructing the state space of the model but rather the alignment of the SCC graph onto time intervals for constructing the HMM states and the exponentiation of rate matrices for computing transition probabilities. The configurations in Table 1 where the construction time is missing were terminated after hours of run-time indicating a very steep exponential growth in running time as the size of the system grows.

4.2 Parameter Estimation

As an evaluation of the model, we consider the so-called *isolation-with-migration* model of speciation, see Fig. 6. In this model, the speciation is initiated by a split in an ancestral species into two groups who evolve independently but exchange genes through limited migration, until at some later point this gene-flow ends and the species evolve completely independent. For most apes, this is believed to be the process in which they separated; the two gorilla species alive today are believed to have split about a million years ago but exchanged genes until recently [29, 31], the two orangutan sub-species has a similar story [17], and even humans have exchanged genes with archaic forms of humans, such as Neanderthals [10] and the recently discovered Denisovan humans [27, 28].

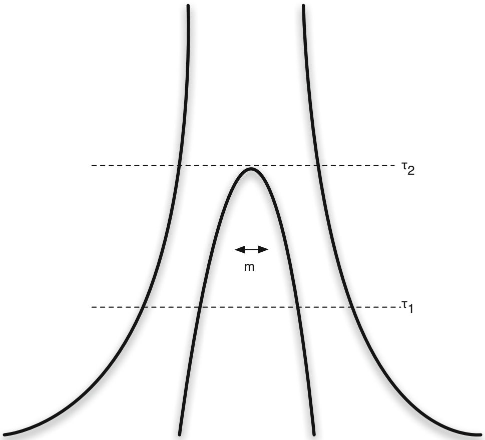


Fig. 6. Isolation-with-migration model. A model of the speciation process, where an ancestral species is initially split into two populations that exchange genes through migration between the groups for a period of time, after which gene flow stops and the species evolve independently.

Relevant parameters in this model include the initial split time, the time when gene-flow ended, the migration rate between the ancestral populations and the coalescence rate in the ancestral species (which measures the genetic variation in the ancestral species). To test our model in this scenario, we simulated data using the coalescence simulator CoaSim [19] and then estimated the parameters with our CoalHMM. Results are shown in Fig. 7.

Although there naturally is some variation in the estimated parameters, we find that the model accurately estimates the parameters of the simulated data.

5 Discussion

We have presented a method for building inference models for complex demographic histories of speciation and genome ancestry. The method 1) employs a colored Petri net to specify the demographic scenario, constructs the state space for the scenario, 2) uses this as a continuous time Markov chain to compute probabilities of genealogies, 3) uses the strongly connected component graph of the state space to compute transition probabilities between local genealogies, and finally 4) uses these transition probabilities to construct a coalescent hidden Markov model for inferring parameters of the scenario.

The approach we have presented fully automates the translation of a demographic scenario to an inference method, making CoalHMMs accessible to biologists with limited computational skills, but the complexity of scenarios that can be explored is still limited by the state space explosion in the model and the computational time needed for enumerating all SCC paths and for exponentiating large rate matrices.

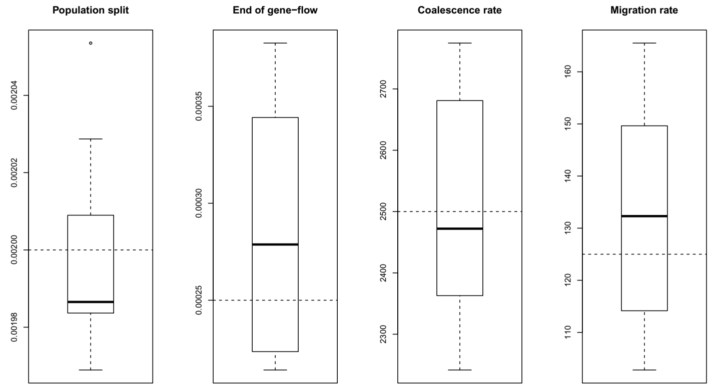


Fig. 7. Parameter estimates on simulated data. In a model with an initial population split followed by a period of migration after which gene-flow stops, we simulated data and estimated parameters. The box plot shows the distribution of estimates in ten simulations. The dashed lines show the simulated parameter.

Different points of attack are possible. The state space explosion can be alleviated using reduction methods. Samples within the same population are by nature symmetric, so symmetry reduction [4, 14] is an obvious approach. The symmetry method allows us to consider two states as equal if it is possible to find a permutation of the samples mapping one to the other. In our example, we can allow all permutations of samples. To evaluate the viability of this approach, we built a prototype implementation of this using CPN Tools [26]. Due to the limitation of our prototype, we have implemented this method using a simple hand-crafted mapping mapping each state to a canonical member of its symmetry group. This means that we not necessarily generate the minimal state space but that we have an efficient means of computing symmetric mappings. In the future we want to either improve this or instead use the lower-level formalism of symmetric nets [2], for which symmetries can be computed automatically. In Table 2, we have shown the reduced sizes of the state space for the different cases. We see that the ratio of the size of the reduced state space to the full state space gets much smaller as the number of species grow.

The sweep-line method [3] makes is possible to delete states from memory when they are no longer needed. In our example, we notice that as soon as two species have coalesced, it is impossible for them to split again. Thus, a state can never have arcs to states where fewer species have coalesced. We can define a progress measure which counting the number of coalesced species and process states in a least progress-first-order. This allows us to delete any states from memory with a lower progress value than any state we have yet to process. We can generate a highly efficient progress measure by creating the state space for the model with only coalescence, computing the strongly connected components of this graph, traverse this graph using a breadth-first traversal and use the

Table 2. Summaries of the state space sizes, SCCs and construction time for the full state space and reduced state spaces. The *basic model* are scenarios with all samples in the same population, while *models with gene-flow* assume one population per sample.

	Full		Symmetry		Sweep-line	Construction time	
Size	States	SCCs	States	SCCs	States	Full	Symmetry
Basic model							
n = 1	2	1	2	1	2	0.00	0.00
n = 2	15	4	12	4	12	0.00	0.00
n = 3	203	25	77	11	46	0.02	0.02
n = 4	4 140	225	607	39	363	0.92	0.30
n = 5	115 975	2 704	5 455	215	2 659	47.05	4.45
n = 6	–	–	54 054	1 604	25 518	–	65.10
n = 7	–	–	586 534	–	266 550	–	2 043.62
Model with gene-flow							
n = 2	94	4	79	4	76	0.01	0.02
n = 3	12 351	25	6 065	10	5 017	4.81	4.99
n = 4	3 188 340	–	731 840	–	451 559	26 525.88	5 720.11

breadth-first rank of each state as progress value for the full state space. In Table 2 we show the maximum number of states in memory during processing (when combined with symmetry reduction). The time to construct the state space and the number of SCCs is the same as for constructing the symmetry reduced graph.

We notice that if we sort the states of the (full) state space such that states belonging to the same strongly connected component are kept together, we get a rate which has block corresponding to each of the strongly connected components. If we know the layout of the blocks we can thus compute the rate matrix from the individual blocks. Each block on the diagonal of the rate matrix corresponds to all transitions internal to a SCC and all other blocks correspond to transitions from one SCC to another. A property of the sweep-line method is that it keeps entire SCCs in memory at the same time (assuming that the progress measure is monotone, which it is here). Thus, we can easily compute all blocks on the diagonal of the rate matrix. Furthermore, we can store, for each SCC, all transitions leading out of the SCC and subsequently sort them according to the target SCC. This allows us to also compute all other blocks. If we sort the states according to the progress measure, we get a rate matrix which is almost upper-triangular. The reason is that it is only possible to go from a state with lesser progress to a state with higher progress. The only parts below the diagonal are the blocks corresponding to transitions internal to a SCC. Furthermore, we know that the rate matrix will be very sparse even above the diagonal, as we only have a non-zero block if there is an arc from one SCC to another.

Unfortunately, the rates of transitions are not necessarily the same even for symmetrical transitions, and it is not obvious to us how to construct the HMM transition matrix from the symmetry reduced CPN state space; future work includes combining lumping techniques for Markov chains [2,6] with the construction of the rate matrix outlines using the sweep-line method to our case.

Rather than exponentiating the rate matrix, it is also possible to get good approximations of the probabilities through Monte Carlo simulation where we can simulate thousands or millions of runs of the continuous time Markov chain and obtain the probabilities this way. Future work will concentrate on ways of extending the complexity of scenarios by alleviating the state space explosion problem.

6 Conclusions

Coalescent hidden Markov models (CoalHMMs) [8,12] are a recent invention that has become popular for genome analysis as they are currently the only approach that is computationally efficient enough for analyzing full genome data. Different variants of CoalHMMs have been successfully used in recent great ape genome projects, including an analysis of human population size changes [16], an analysis of the orangutan sub-species [17,18], and estimating speciation times between humans, chimpanzees, bonobos, gorillas and orangutans [13,25,29]. The demographic scenarios explored, however, have been very simple ones because the CoalHMMs have so far been constructed manually. Typically, this involves deriving equations for transition probabilities by approximating the coalescence process, which at best is tedious and in cases can introduce biases in the estimation because of the simplifying assumptions necessary to do this. Computing the transition probabilities using a continuous time Markov chain alleviates this somewhat, but manually constructing the Markov chain is still only possible for simple scenarios.

While the colored Petri net model we present here is very simple, we stress that it is capable of modeling most demographic scenarios. Combined with an algorithm for translating a formal model of demographics like this into the final CoalHMM, complex scenarios can be explored in genome analysis. As the cost of sequencing genomes is steadily decreasing, the bottleneck in future genome projects will be in the mathematical modeling and in constructing analysis methods that both captures the complexity of the genomes and are computationally efficient. We believe that CoalHMMs combined with formal methods such as Petri nets can be a powerful approach in this.

Acknowledgements. The authors wish to thank Lars M. Kristensen for fruitful discussions about the use state-space reduction techniques.

References

1. Chen, G.K., Marjoram, P., Wall, J.D.: Fast and flexible simulation of DNA sequence data. Genome Res. 19(1), 136–142 (2009)

2. Chiola, G., Dutheillet, C., Franceshinis, G., Haddad, S.: Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications. IEEE Trans. Computers 42(11), 1343–1360 (1993)
3. Christensen, S., Kristensen, L.M., Mailund, T.: A Sweep-Line Method for State Space Exploration. In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 450–464. Springer, Heidelberg (2001)
4. Clarke, E., Emerson, E., Jha, S., Sistla, A.P.: Symmetry Reductions in Model Checking. In: Vardi, M.Y. (ed.) CAV 1998. LNCS, vol. 1427, pp. 147–158. Springer, Heidelberg (1998)
5. Davison, D., Pritchard, J.K., Coop, G.: An approximate likelihood for genetic data under a model with recombination and population splitting. Theoretical Population Biology 75(4), 331–345 (2009)
6. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in markov chains. Inf. Process. Lett. 87(6), 309–315 (2003)
7. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids. Cambridge Univ. Pr. (February 2005)
8. Dutheil, J.Y., Ganapathy, G., Hobolth, A., Mailund, T., Uyenoyama, M.K., Schierup, M.H.: Ancestral population genomics: the coalescent hidden Markov model approach. Genetics 183(1), 259–274 (2009)
9. Eriksson, A., Mahjani, B., Mehlig, B.: Sequential Markov coalescent algorithms for population models with demographic structure. Theor. Popul. Biol. 76(2), 84–91 (2009)
10. Green, R.E., et al.: A draft sequence of the neandertal genome. Science 328(5979), 710–722 (2010)
11. Hein, J., Schierup, M.H., Wiuf, C.: Gene genealogies, variation and evolution. a primer in coalescent theory. Oxford University Press, USA (2005)
12. Hobolth, A., Christensen, O.F., Mailund, T., Schierup, M.H.: Genomic relationships and speciation times of human, chimpanzee, and gorilla inferred from a coalescent hidden Markov model. PLoS Genet 3(2), e7 (2007)
13. Hobolth, A., Dutheil, J.Y., Hawks, J., Schierup, M.H., Mailund, T.: Incomplete lineage sorting patterns among human, chimpanzee, and orangutan suggest recent orangutan speciation and widespread selection. Genome Res. 21(3), 349–356 (2011)
14. Jensen, K.: Condensed State Spaces for Symmetrical Coloured Petri Nets. Formal Methods in System Design 9(1/2), 7–40 (1996)
15. Jensen, K., Kristensen, L.M.: Coloured Petri Nets. Modeling and Validation of Concurrent Systems. Springer-Verlag New York Inc. (June 2009)
16. Li, H., Durbin, R.: Inference of human population history from individual whole-genome sequences. Nature (July 2011)
17. Locke, D.P., et al.: Comparative and demographic analysis of orang-utan genomes. Nature 469(7331), 529–533 (2011)
18. Mailund, T., Dutheil, J.Y., Hobolth, A., Lunter, G., Schierup, M.H.: Estimating Divergence Time and Ancestral Effective Population Size of Bornean and Sumatran Orangutan Subspecies Using a Coalescent Hidden Markov Model. PLoS Genet. 7(3), e1001319 (2011)
19. Mailund, T., Schierup, M.H., Pedersen, C.N.S., Mechlenborg, P.J.M., Madsen, J.N., Schauser, L.: CoaSim: a flexible environment for simulating genetic data under coalescent models. BMC Bioinformatics 6, 252 (2005)
20. Marjoram, P., Wall, J.D.: Fast “coalescent” simulation. BMC Genetics 7, 16 (2006)
21. Marsan, M.: Stochastic Petri Nets: An Elementary Introduction. In: Rozenberg, G. (ed.) APN 1989. LNCS, vol. 424, pp. 1–29. Springer, Heidelberg (1990)

22. McVean, G.A.T., Cardin, N.J.: Approximating the coalescent with recombination. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 360(1459), 1387–1393 (2005)
23. Moler, C., van Loan, C.: Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. SIAM Review 45(1), 3–49 (2003)
24. Paul, J.S., Steinrucken, M., Song, Y.S.: An Accurate Sequentially Markov Conditional Sampling Distribution for the Coalescent With Recombination. Genetics 187(4), 1115–1128 (2011)
25. Prüfer, K., et al.: The bonobo genome compared with the genomes of chimpanzee and human, under review at Nature
26. Vinter Ratzer, A., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 450–462. Springer, Heidelberg (2003)
27. Reich, D., et al.: Genetic history of an archaic hominin group from denisova cave in siberia. Nature 468(7327), 1053–1060 (2010)
28. Reich, D., et al.: Denisova admixture and the first modern human dispersals into southeast asia and oceania. Am. J. Hum. Genet. 89(4), 516–528 (2011)
29. Scally, A., et al.: Insights into hominid evolution from the gorilla genome sequence. Nature 483(7388), 169–175 (2012)
30. Song, Y.S., Lyngso, R., Hein, J.: Counting All Possible Ancestral Configurations of Sample Sequences in Population Genetics. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 3(3), 239 (2006)
31. Thalmann, O., Fischer, A., Lankester, F., Pääbo, S., Vigilant, L.: The complex evolutionary history of gorillas: insights from genomic data. Mol. Biol. Evol. 24(1), 146–158 (2007)

A Hidden Markov Models

Hidden Markov models (HMMs) is a framework for modeling sequential data such as DNA sequences. HMMs provide a computational efficient way of analyzing sequential data, that would otherwise be intractable to analyze.

Given a sequence of observation $Z_1, Z_2, \ldots, Z_n$, the Markov assumption states the the probability of the entire sequence $\Pr(Z_1, Z_2, \ldots, Z_n)$ can be stated as a sequence of conditional probabilities

$$\Pr(Z_1, Z_2, \ldots, Z_n) = \Pr(Z_1) \prod_{i=1}^{n-1} \Pr(Z_{i+1} \mid Z_i)$$

which is typically much more efficient to calculate.

In many applications, however, a sequence of observations cannot be justified modeled in this way. The genetic differences between a sample of genes, for instance, is conditional on the local genealogies, but whereas we can model the local genealogies as a Markov process the observed genetic differences do not directly lead to a Markov process.

Hidden Markov models instead assumes that we have a sequence of unseen parameters that *do* follow a Markov process, but that the observations we see depend on those parameters but are not themselves a Markov process. An HMM models a sequence of observations, $X_1, X_2, \ldots, X_n$, by assuming there is an underlying but unobserved sequence of states the process goes through, $Z_1, Z_2, \ldots, Z_n$, that determines the probability of the observations. Each observation depends on one hidden state, $\Pr(X_i \mid X_1, \ldots, X_n, Z_1, \ldots, Z_n) = \Pr(X_i \mid Z_i)$, as the genetic differences between a set of genes would depend on the local genealogies at each position but not neighboring genealogies. Were both the hidden states and the observations known, the joint probability would be

$$\Pr\left(\mathbf{X}, \mathbf{Z}\right) = \Pr\left(Z_1\right) \Pr\left(X_1 \mid Z_1\right) \prod_{i=1}^{n-1} \Pr\left(Z_{i+1} \mid Z_i\right) \Pr\left(X_{i+1} \mid Z_{i+1}\right) \quad .$$

The Markov process states, Z_i, however are not observed in a *hidden* Markov model, only the sequence $X_1, \ldots, X_n$. Efficient dynamic programming algorithms exist, however to sum over all hidden state paths and thus computing $\Pr(\mathbf{X})$ and from this making maximum likelihood parameter estimations.

An HMM is completely parameterized by specifying the initial state probabilities $\pi = (\Pr(Z_1), \ldots, \Pr(Z_m))$ for possible hidden states $Z_1, \ldots, Z_m$, the *transition* probability matrix $T_{i,j} = \Pr(Z_i \mid Z_j)$ and the *emission* probability matrix $E_{l,m} = \Pr(X_m \mid Z_l)$.