
FROM EXACT MARGINALS
TO BETTER IMPORTANCE SAMPLING

SEPTEMBER, 2008

CAMILLA MONDRUP ANDREASSEN

&

ANDERS HAUGE OKHOLM

Contents

1	Introduction	1
1.1	Genes and alleles	1
2	Infinite Sites Model (ISM)	1
3	Configuration networks	2
4	Representation of data	4
5	Calculating the probability	4
5.1	Likelihoodcurve	9
6	Importance Sampling	10
6.1	Effective Sample Size (ESS)	12
7	Previous proposal distributions	13
7.1	Stephens-Donnelly proposal	13
7.2	Hobolth proposal	14
8	The new proposal	15
9	Results	16
9.1	Calculations based on data-set A	16
9.2	Calculations based on data-set B	19
10	Discussion and conclusion	20
	Discussion - How to improve our proposal	20
	Conclusion	22
11	Aknowledgements	22
	Literature	23

1 Introduction

In the fields of biology and medicine, gene evolution has been of great interest for researchers the past 50 years. Ever since it has been possible to isolate and examine gene populations, knowledge of their evolution has been more and more significant.

DNA sequences from a population have an unobservable genealogical history. In analyzing such data, a crucial stepping-stone is to be able to integrate over evolutionary histories according their probability according to a model and given the data. Doing this has been the focus of research for more than 2 decades

This report is on optimizing the estimate of the probability of observing a given gene population. The report will be in two parts. First part will be an introductory to the terminology and mathematical models. Second part will introduce previous optimizing proposals, which will lead us to our new proposal. Our proposal is based on the idea that analysing sub-sets of a data-set will improve the calculation of the probability.

1.1 Genes and alleles

The way a person looks or appears is determined by their genes. Genes are segments of DNA, containing the information about a specific characteristic, for example eye or hair colour. There exists different types of each gene, and each type may result in different traits, i.e. what is the colour of the eyes or the hair.

Different types of a gene are called alleles of that gene. A type of gene for eye colour resulting in blue eyes is one allele of the gene, while a type of gene resulting in brown eyes is another.

What kind of information the allele carries depends on the nucleotide sequence of the DNA. Mutations are random events changing the sequence of the gene (e.g. replacement of an adenine by a guanine), resulting in a new allele. Mutations can for example be caused by copying errors in the genetic material during cell division or exposure to ultraviolet light.

2 Infinites Sites Model (ISM)

ISM is a model to describe the evolution of a very long DNA sequence with low mutation rate at each nucleotide. Due to this, we can neglect the possibility of two or more mutations happening on the same site. For that reason we can represent the sequences by 0's and 1's, where 0 is an unchanged site and 1 is where mutation has occurred. The model is reasonable because we usually observe few variable sites

in a sample of sequences. In the model we neglect recombination and any other kind of change in the gene-structure. We only consider mutation and duplication.

3 Configuration networks

A configuration network is a model that shows all possible ancestral states, connected in a compatible order. Configuration networks give a quick view of the histories for a data-set. A history is a connected path up through the configuration network, determined by the states it passes. The following will be an accessible introduction to configuration networks and how to generate them.

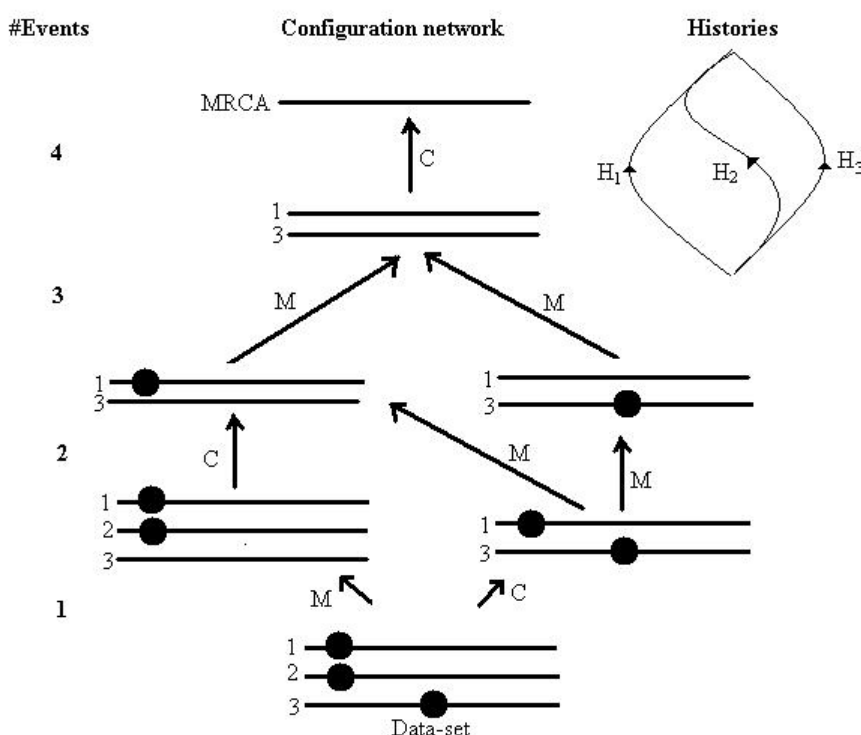


Figure 1: Configuration network of toy dataset.

Look at figure 1. Every horizontal line represents an allele and the black balls represent mutations on the alleles. The numbers in the left column are the numbers of events from the bottom configuration. It will later be obvious why we label the events from bottom to top. An event describes in our case either a mutation or a duplication of the allele. The arrows in the figure indicate such events, backward in time though. The time passes while going down the configuration network. Imagine, that if we could follow the evolution of a specific allele, we would know whether the first, second and third event were respectively duplication- or mutation-events. We would then know the actual history down the configuration network. But what we

can observe now, is only the last configuration in the bottom, which we will refer to as the data-set. In this case, the data-set consists of three alleles, labelled 1-3. 1 and 2 contains a mutation on the same segregating site, while 3 has got a mutation on a different site. The top allele labelled MRCA is the most recent common ancestor for the data-set. The configuration for MRCA is also known.

From knowledge of the data-set, we would like to build up an ancestral tree containing all possible histories back to MRCA. As the figure should tell, there are three possible ways (histories). The histories are shown to the right of the figure. Normally, only having the data-set and the common ancestor, we will have to build up the configuration network backwards in time considering the possible events. As the data-set grows larger, one could easily imagine that the numbers of histories grows exponentially, and the configuration network is no longer trivial to draw. Despite this, for simple data-sets as the one in figure 1, we can draw the configuration network. The following will be a guide of how to build up configuration networks, given a data-set.

The two events to consider are coalescence (C) and removal of a mutation (M). Coalescence is simply the opposite of a duplication that is a fusion of two identical alleles. We have to introduce these terms, recording that we are moving backward in time as these events occur. Coalescence can happen, whenever two or more alleles are identical in a configuration. A mutation can only be removed if it is the only mutation on its site. This is a consequence of our assumption that mutations only happens once on a site, from the ISM. This implies, that if two alleles have the same mutation, they must first coalesce, before the mutation can be removed.

In the example in figure 1, we observe, that the data-set contains two identical alleles (1 and 2) and a unique allele (3) with a single site. The last event before the data-set-configuration could either have been a coalescence event (fusion of 1 and 2) or removal of a mutation (3). The arrows pointing from the data-set to previous ancestral configurations are marked by C or M, telling what event just occurred. By this first step up the event-latter, we have already generated two possible histories. This process continues as just described, by considering the latest possible events for each configuration. Generally it is helpful to focus on all configurations before proceeding to the next step on the event-latter. Notice that two identical configurations cannot appear in the tree. Whenever two different descendants have the same ancestor, the two arrows should lead to the same ancestor. It is illustrated when second event occurs. The left configuration experiences coalescence, and if the configuration to the right experiences a removal of a mutation they become the same. Finally you end up with two alleles identical to the MRCA, and when these coalesce you have your configuration network.

4 Representation of data

It was necessary to represent the data-configurations in matlab to do calculations on the histories. To represent our data-set in matlab we used binary numbers in a matrix \mathbf{S} with dimensions $k \times m$. k is the number of different alleles, and m is the number of segregating sites with at least one mutation, in the data-set. In reference to ISM every entry in the matrix is either 0 or 1. The entry $\mathbf{S}_{ij} = 0$ if the allele i does not have a mutation on the segregating site j . Otherwise if $\mathbf{S}_{ij} = 1$ i has a mutation on j . Furthermore we introduce the column vector \mathbf{N} , which gives the multiplicity of each type of allele in the data-set. The dimension of \mathbf{N} is $k \times 1$. If the data-set contains three identical alleles, and they are represented with the row i in \mathbf{S} , the entry i in \mathbf{N} will be 3. Recall the data-set from figure 1 on page 2.

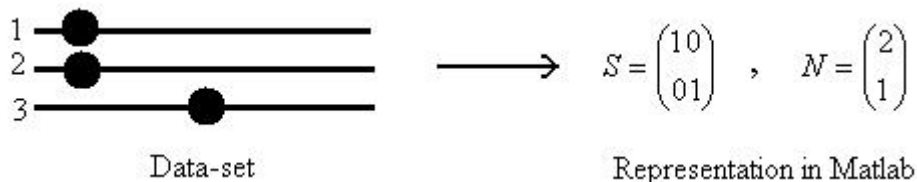


Figure 2: Representation of toy data-set.

We chose to represent configurations in this way, because it allows us to do the relevant operations and is a compact form of the information. By looking at the representation we are able to tell what the next event could be for the actual configuration. This representation is also easy to update.

5 Calculating the probability

Before jumping right into probability calculations, we have to introduce the value θ called the scaled mutation rate. θ is just referred to as the mutation rate, and it is mathematically defined as $\theta = 4N\mu$, where μ is the mutation rate per generation, and N is the population size. Our data-set depends on θ . A data-set generated with a relatively low θ , will be more likely to contain few mutations compared to data-sets generated with large values of θ . Having introduced this parameter we are now ready to focus on the true essence of this report; calculating the likelihood of a data-set.

We want to calculate the probability $P(D|\theta)$ of observing a data-set D if the mutation rate take the value θ , and if we know the MRCA of D . The likelihood $L(\theta)$ is a function of θ , defined to take the quantity $P(D|\theta)$.

$$L(\theta) = P(D|\theta)$$

P_C , n_j is the multiplicity of the gene chosen to coalesce. With that said, it should be obvious that $n_j > 1$. In the term for P_M , n_i expresses the multiplicity of the gene, which the gene that is chosen to have a mutation removed, reduces to. If the gene is still unique after removal of a mutation, $n_i = 0$, but if the gene after having a mutation removed is suddenly identical to three other genes, $n_i = 3$. The first factor in P_C and P_M are the probabilities of coalescence and mutation event. The second factors are the probabilities of choosing the specific type of gene which is target of the event, this is simply just $\#\{succes\}/\#\{possible\}$. We will go through an example where we calculate the probabilities of two possible events. As a starting point we will use A from figure 3, calculating the probabilities of going to either A_{d1} or A_{d2} for $\theta = 1.22$.

Probability of $A \rightarrow A_{d1}$: To go from A to A_{d1} , gene 1 needs to duplicate. Therefore we only have to consider P_C . $n = 3$ because the total amount of genes in A_{d1} is three. $n_j = 2$ because the number of gene 1 in A_{d1} is two.

$$P(A_{d1}|A) = \frac{n-1}{n-1+\theta} \cdot \frac{n_j-1}{n-1} = \frac{3-1}{3-1+1.22} \cdot \frac{2-1}{3-1} = \frac{1}{3.22} = 0.311$$

Probability of $A \rightarrow A_{d2}$: To go from A to A_{d2} , gene 3 needs to get a mutation. Therefore we only have to consider P_M . $n = 2$ because the total amount of genes in A_{d2} is two. $n_i = 0$ because the number of genes in A which are identical to gene 3 in A_{d2} is 0.

$$P(A_{d2}|A) = \frac{\theta}{n-1+\theta} \cdot \frac{n_i+1}{n} = \frac{1.22}{2-1+1.22} \cdot \frac{0+1}{2} = \frac{1.22}{2.22} \cdot \frac{1}{2} = 0.275$$

We will not care about the events following the dashed lines. But they give us a problem to consider. We need to work out a model that only takes possible events into concern, and then calculate their possibilities. In figure 3, we showed, that going one event forward in time for A , would give us four descendants, two of them incompatible with our data-set. On the other hand, if we do our calculations backwards, that is starting with the data-set in the bottom, then moving up the configuration network to every possible state just one event away, we will avoid calculating histories incompatible with our data-set. For every move up the tree, the previous configuration will contain enough information to decide whether an event is possible or not. The calculation of probabilities is analogous to the procedure of drawing the configuration network. Both based on the fact that only compatible ancestral states can be deduced from its descendants.

When we can calculate the possibility $P(C_g|C_{g-1})$ of going from one configuration in generation $g - 1$ to another configuration in generation g just one event away, we can easily calculate the probability $P(\mathcal{H})$ for a history \mathcal{H} . $P(\mathcal{H})$ is simply the product of $P(C_g|C_{g-1})$ for all the involving steps in the history. The probability of a history for a data-set, where there are G generations between data-set and MRCA, is

$$P(\mathcal{H}) = \prod_{g=2}^G P(C_g|C_{g-1}) \quad (3)$$

The likelihood of the data is the sum of all possible histories leading from MRCA to the data. If \mathcal{H}_b is one history out of B compatible histories, the likelihood is

$$L(\theta) = P(D|\theta) = \sum_{b=1}^B P(\mathcal{H}_b) \quad (4)$$

If this conclusion does not seem trivial to you, recall the definition of the likelihood of the data.

For data sets, as the one in figure 1, we can by hand calculate $L(\theta)$, by a straightforward method, calculating all $P(C_g|C_{g-1})$, and all $P(\mathcal{H}_b)$. Even for the very simple data-sets like this, it takes quite a while to calculate all these values. To cope with this problem it is possible to derive a recursion. When implemented in e.g. Matlab, the recursion can calculate the probability a lot faster. We calculated the likelihood of the data-set from figure 1, with $\theta = 1.22$. $L(1.22) = 0.0520$ and the elapsed time was 0.078 s.

The recursion is

$$\begin{aligned} L(\theta) = P(\mathbf{S}, \mathbf{N}|\theta) &= \frac{n-1}{n-1+\theta} \sum_{j:n_j>1} \frac{n_j-1}{n-1} P(\mathbf{S}, \mathbf{N}_1 - \mathbf{e}_j|\theta) \\ &+ \frac{\theta}{n-1+\theta} \sum_{j:\text{singletons}} \frac{n_i+1}{n} P(\mathbf{S}^\sim, \mathbf{N}^\sim|\theta) \end{aligned} \quad (5)$$

Here follows an explanation to the recursion. The recursion calculates the full likelihood for the data-set. \mathbf{S} is a matrix and \mathbf{N} is a vector uniquely representing the data-set. \mathbf{S}^\sim and \mathbf{N}^\sim represent the ancestor configuration after mutation. Understand $P(\mathbf{S}, \mathbf{N}|\theta)$ as $P(D|\theta)$. In the recursion there should be a lot of familiar terms. New things introduced, are the summation over the second factor in P_C and P_M . This summation just makes sure that we include all possibilities for our data-set, recall (4). There might be more than one kind of genes which can coalesce in our data-set. And there might as well be more than one unique gene which can mutate. Figure 4 on the next page shows a data-set for which the recent event can be either one of the five shown.

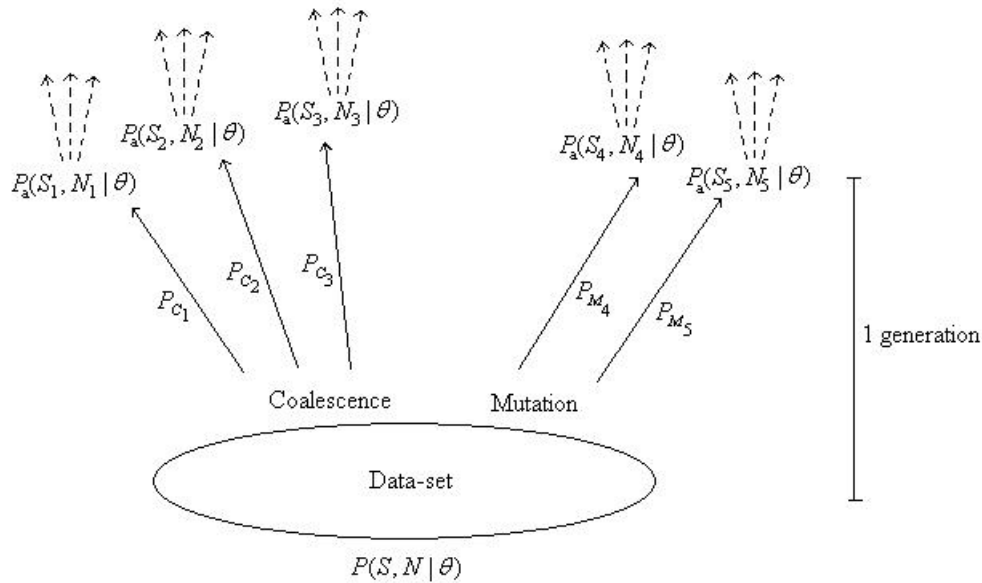


Figure 4

Three different coalescence events and two different mutation events. Recalling (3), we need all the values for P_{C_1} , P_{C_2} , P_{C_3} , P_{M_4} and P_{M_5} to calculate the likelihood of the data. The five steps each generate their own unique histories. So $P(\mathbf{S}, \mathbf{N}|\theta)$ gives the sum of all the probabilities from the first step multiplied with the probability of the rest of the history, according to (3) and (4). The first step in the recursion will look like this:

$$P(\mathbf{S}, \mathbf{N}|\theta) = P_{C_1} * P_a(\mathbf{S}_1, \mathbf{N}_1|\theta) + P_{C_2} * P_a(\mathbf{S}_2, \mathbf{N}_2|\theta) + P_{C_3} * P_a(\mathbf{S}_3, \mathbf{N}_3|\theta) \\ + P_{M_4} * P_a(\mathbf{S}_4, \mathbf{N}_4|\theta) + P_{M_5} * P_a(\mathbf{S}_5, \mathbf{N}_5|\theta)$$

Now $\mathbf{S}_n, \mathbf{N}_n$ represents a new configuration after the first step up the network. $P_a(\mathbf{S}_n, \mathbf{N}_n)$ is the likelihood of the sub-data-set appearing. $\mathbf{S}_n, \mathbf{N}_n$ is then run through the recursion as if it were the actual data-set of interest. This will generate the new probable paths, and as one would start to think, the recursion branches out all the possible histories. The same procedure follows, and finally we will end up with just as many terms as there are histories, each with the same amount of factors, that is the total number of events from the data-set to the MRCA. We know all histories start at the MRCA, so this configuration will have probability 1. That is why $P(\mathbf{S}^*, \mathbf{N}^*|\theta) = 1$ where $\mathbf{S}^*, \mathbf{N}^*$ represents the MRCA. In figure 5 on the following page we have shown how the recursion works on our toy data-set step by step.

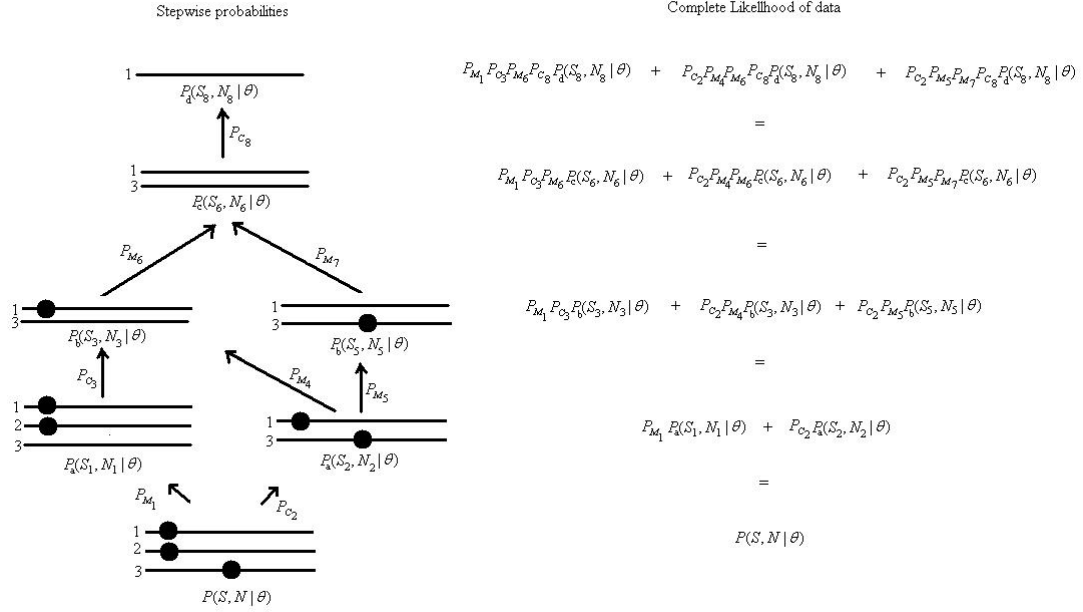


Figure 5

Remember that the stepwise probabilities are determined by (1) and (2) and that $P_d(\mathbf{S}_8, \mathbf{N}_8 | \theta) = 1$. Both graphs should be read from bottom to top. The left graph is the configuration network. Between two configurations are placed the probability of the upper configuration evolving into the configuration underneath. Note that this is the opposite direction to the arrows. The probabilities placed just underneath each configuration are the probabilities of observing these configurations, knowing MRCA. The right graph shows how the recursion runs, following the same steps as the left graph. All the equations are equal at each step. The graph is only to illustrate how the recursion is unfolded.

5.1 Likelihoodcurve

By use of the recursion, we can calculate $P(D|\theta)$, that is the likelihood of the data. If we plot the likelihood $P(D|\theta)$ against θ we will get the likelihood curve. By maximizing the function, we will get the value, which was most likely to generate the dataset. If we consider the dataset from figure 1:

$$\mathbf{S} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{N} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad (6)$$

we will get the curve illustrated in figure 6 on the next page. The curve has a maximum at approximately 1.22, and thereby we get, that $\theta = 1.22$ is the most

likely mutation rate.

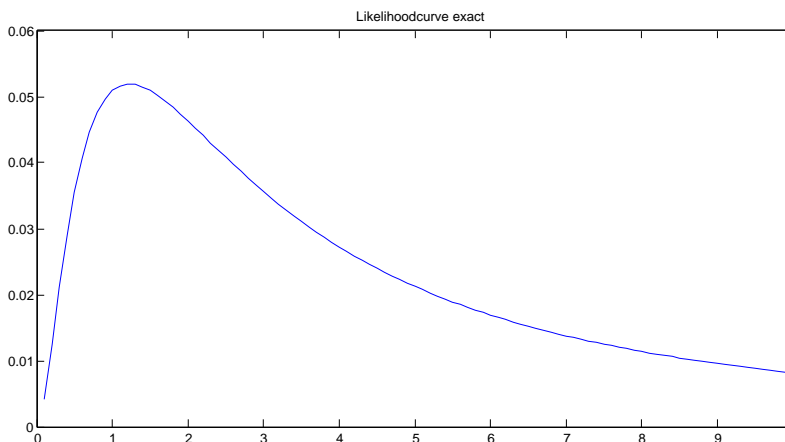


Figure 6: Exact likelihood curve for the dataset (6).

We have now given an introduction to calculate the likelihood of the data exact. The recursion introduced in this chapter is the total solution for any data set when calculating the likelihood in the Infinite sites model. But as data-sets become larger and the number of segregating sites increases, computations with the recursion on the whole data-set become useless. The problem is, that usual data-sets examined in the light of science falls under the category too complex. We simply have not got the computer power to make the calculation in a reasonable amount of time. This fact induces the great importance of approximating the likelihood. The quality of an estimate is defined by its accuracy in determining the likelihood and by the time it takes. To do the best approximation to the likelihood has been a discipline of interest since Griffiths and Tavaré faced the problem in 1994. Since that, the most common approach to the problem is using a technique called Importance sampling.

6 Importance Sampling

We already mentioned that

$$L(\theta) = P(D|\theta) = \sum_{b=1}^B P(\mathcal{H}_b)$$

and we can rewrite this by multiplying and dividing with the probability $Q(\mathcal{H})$. By definition, the new term is the expectation over $Q(\mathcal{H})$ of $P(\mathcal{H})/Q(\mathcal{H})$. We get

$$L(\theta) = P(D|\theta) = \sum_{b=1}^B P(\mathcal{H}_b)Q(\mathcal{H}_b)/Q(\mathcal{H}_b) = E_Q[P(\mathcal{H})/Q(\mathcal{H})] \quad (7)$$

Q is a proposal distribution of the probability of a history from the date-set to the MRCA. Remember that $P(\mathcal{H})$ is the actual probability of a history from the MRCA to the Data-set, that is P and Q works in opposite directions.

Importance sampling is based on the approximation

$$\begin{aligned} L(\theta) &= P(D|\theta) = \sum_{b=1}^B P(\mathcal{H}_b)Q(\mathcal{H}_b)/Q(\mathcal{H}_b) \\ &= \sum_{\mathcal{H}} P_{\theta}(\text{Data}, \mathcal{H})Q(\mathcal{H})/Q(\mathcal{H}) = E_Q[P_{\theta}(\text{Data}, \mathcal{H})/Q(\mathcal{H})] \\ &\approx \frac{1}{R} \sum_{r=1}^R P_{\theta}(\text{Data}, \mathcal{H}_r)/Q(\mathcal{H}_r) = \frac{1}{R} \sum_{r=1}^R w_r \end{aligned}$$

This approximation is justified by the law of large numbers. This equation states, that the ratio of $P(\mathcal{H})$ and $Q(\mathcal{H})$, also referred to as the weight w , is an estimate of $L(\theta)$. So sampling one history will give

$$P(D|\theta) \approx P(\mathcal{H}_b)/Q(\mathcal{H}_b) = w_b \quad (8)$$

If we sample a lot of histories, calculating w for all of them and then taking the average of w , we will get a good estimate for $P(D|\theta)$. So whenever we explore one possible history, we will get an estimate for $P(D|\theta)$. Remember, that when we calculated $P(D|\theta)$ exact with the recursion we had to run through all possible histories. So using this technique is an efficient timesaver.

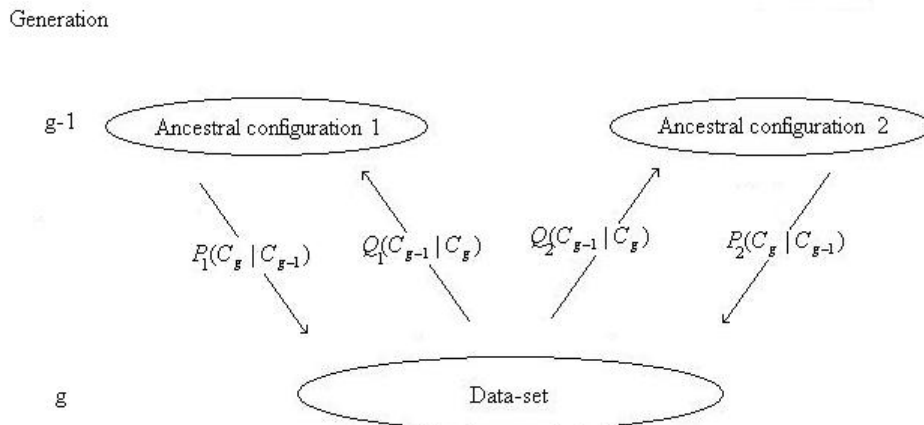


Figure 7

The procedure for calculating w_b is illustrated in figure 7. The figure shows two moves back in time from one configuration C_g to another C_{g-1} in the sampled histories. According to (3), (8) can be written as

$$P(\mathcal{H}_b)/Q(\mathcal{H}_b) = \prod \frac{P(C_g|C_{g-1})}{Q(C_{g-1}|C_g)}$$

We have shown before how to easily calculate $P(C_g|C_{g-1})$ by either (1) or (2). $Q(C_{g-1}|C_g)$ on the other hand is unknown, that is why Q is called the proposal distribution. The optimal $Q(\mathcal{H})$ would be

$$Q(\mathcal{H})_{\text{opt}} = \frac{P(\mathcal{H})}{P(D|\theta)}$$

Inserted in (8), we realise that this is an ideal situation, where we only have to do one sample. Running through any given history with $Q(\mathcal{H})_{\text{opt}}$ will immediately give us $w_b = P(D|\theta)$.

The challenge in importance sampling is to choose the proposal distribution Q well. For every proposal distribution, however bad it would be, the average of P/Q will eventually converge to $P(D|\theta)$, one will just have to sample an enormously amount of histories. But if a good Q is chosen, the convergence will happen a lot faster.

One thing we know for sure is that, going one event back in time from a configuration, the sum over all $Q(C_{g-1}|C_g)$ is 1. Because as we stated earlier, all possible ways to go backward in time are consistent with the data-set, and therefore the possibilities of going these ways must sum to 1. Whenever Q does not sum to 1 we normalise it.

Notice that if $Q(\mathcal{H})$ over proposes a history \mathcal{H} , then the importance sampler will choose this path more often than it should, resulting in w_b underestimates $P(D|\theta)$. Otherwise if our probability proposal of a history is less than it should be, w_b overestimates $P(D|\theta)$.

6.1 Effective Sample Size (ESS)

A way to tell if the new proposal is better than the previous is to calculate the Effective Sample Size ESS (Liu, 2001, Section 2.5). It is defined as

$$ESS = \frac{R}{1 + \left(\frac{\sigma}{\mu}\right)^2} \quad (9)$$

where R is the number of samples, σ is the standard deviation and μ is the mean of the weights.

A usefull estimate of the ESS is given by

$$E\hat{SS} = \frac{(\sum weights)^2}{\sum weights^2} \quad (10)$$

7 Previous proposal distributions

In the following we will introduce two previous proposals for an importance sampler to estimate the likelihood of a data-set.

7.1 Stephens-Donnelly proposal

The SD proposal is quite simple and a straightforward definition of Q . It is based on calculations in the IAM model and the allele is simply just chosen uniformly at random.

Let k be an allele in the configuration. \mathbb{M} is the set of alleles corresponding to a singlet site - alleles that are able to undergo mutation. If \mathbf{N}_k is the multiplicity of an allele, then

$$Q^{SD}(k|\mathbf{S}, \mathbf{N}) \propto \begin{cases} \mathbf{N}_k & \text{if } \mathbf{N}_k \geq 2 \text{ or } k \in \mathbb{M} \\ 0 & \text{if } \mathbf{N}_k = 1 \text{ and } k \notin \mathbb{M} \end{cases}$$

It means that if allele k cannot be the target of the next event back in time, $Q = 0$. Otherwise if k can be the target of the next event, Q is proportional to the amount of identical alleles to k . Notice, that the proposal does not include a driving value for θ .

For our toy data-set (6), Q^{SD} will be

$$Q^{SD}(k|\mathbf{S}, \mathbf{N}) \propto \begin{cases} 2 & \text{if } k = 1 \\ 1 & \text{if } k = 2 \end{cases}$$

and thereby

$$Q^{SD}(k|\mathbf{S}, \mathbf{N}) = \begin{cases} 2/3 & \text{if } k = 1 \\ 1/3 & \text{if } k = 2 \end{cases}$$

The SD proposal foresees, that 1 and 2 will coalesce with probability 2/3 and that 3 will mutate with probability 1/3.

SD does not take the probability of the ancestral state into concern. As for this example, Q^{SD} indicates that it is more likely to coalesce 1 and 2, than to mutate 3. But it might be, that the configuration appearing after coalescing 1 and 2 is much more unlikely than the one appearing after mutation of 3. It means that Q^{SD} in some cases proposes unlikely paths with a high probability. We cannot neglect the probability of the ancestral states Q leads us to.

7.2 Hobolth proposal

The Hobolth proposal takes the ancestral states into account. Hobolth's idea is to isolate the individual sites in data-sets. By looking at only one site at a time, there is only two possible states of each allele, either mutation or not. No matter how many alleles we have got, all individual sites can be represented by

$$\mathbf{S} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } \mathbf{N} = \begin{pmatrix} d_m \\ n - d_m \end{pmatrix}$$

with $d_m = \sum_k \mathbf{S}_{km} N_k$ being the number of alleles with a mutation. Hobolth has derived an equation that gives us the probability p_θ , that the next event backward in time reduces the number of genes with the mutation by one. The probability $1 - p_\theta$ is then the probability of reducing the number of alleles without a mutation by one. p_θ is given by

$$\begin{aligned} p_\theta(d) &= P \left(\mathbf{S} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} d-1 \\ n-d \end{pmatrix} \middle| \mathbf{S} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} d \\ n-d \end{pmatrix} \right) \\ &= \frac{\sum_{k=2}^{n-d+1} \frac{d-1}{n-k} \frac{1}{k-1+\theta} \binom{n-d-1}{k-2} \binom{n-1}{k-1}^{-1}}{\sum_{k_0=2}^{n-d+1} \frac{1}{k_0-1+\theta} \binom{n-d-1}{k_0-2} \binom{n-1}{k_0-1}^{-1}}. \end{aligned}$$

With this knowledge Hobolth could calculate the backward probability for a data-set with only one segregating site. To fit real size data-sets he applied this easy soluble model to the big data-sets. Hobolth simply splitted the data-sets up in subsequences all with only 1 segregating site.

Imagine a data-set with 18 segregating sites and 30 alleles. Using the Hobolth method we would analyze the 18 sites separately, and for each site, we will calculate the probability for each of the alleles to be the next target of an event. So for each of the 30 alleles we get 18 probabilities of it being the next target of an event. Now, for each allele, Hobolth sum the probabilities over all the subsequences. The probability that an allele is the target of the next event in the perspective of the whole data-set is proportional to the sum of all the probabilities, u_m , of this allele being target of the next event for each segregating site.

$$u_{km}(\theta) = \begin{cases} p_\theta(d_m) N_k / d_m & \text{if } \mathbf{S}_{km} = 1 \\ (1 - p_\theta(d_m)) N_k / (n - d_m) & \text{if } \mathbf{S}_{km} = 0 \end{cases}$$

The concept of the Hobolth proposal, is to agree that we can only solve simple problems, and then split the complex problem into such simple problems. The tricky part is to combine all the simple solutions, to get a solution for the complex

problem. Hobolths proposal:

$$Q_{\theta_0}^{Hob}(k|\mathbf{S}, \mathbf{N}) \propto \begin{cases} \sum_m u_{km}(\theta_0) & \text{if } N_k \geq 2 \text{ or } k \in \mathbb{M} \\ 0 & \text{if } N_k = 1 \text{ and } k \notin \mathbb{M} \end{cases}.$$

8 The new proposal

We are now ready to introduce our new proposal. It is based on Hobolths idea of splitting a complex problem into small soluble problems, and Bayes' Theorem.

To expand Hobolths idea, we will not only look at one site at a time, but cut out subsequences of two or more sites.

$$\mathbf{S} = \begin{pmatrix} \dots & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ \dots & 0 & 0 & 1 & 0 & 0 & 1 & \dots \\ \dots & 1 & 1 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \longrightarrow \dots \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \dots$$

The subsequences were constructed randomly by picking random sites and joining them to a subsequence.

For each subsequences we will calculate the backward probability corresponding to Hobolths p_θ . We will examine if this leads to any improvement in estimating the likelihood of the data, and also consider what the optimal choice of size of the subsequences is.

For each subsequence we make a new representation. Let \mathbf{S}_m be a subsequence.

$$\mathbf{S}_m = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{N}_m = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \longrightarrow \mathbf{s}_m = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{n}_m = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

To calculate the backward probability, we will use Bayes' Theorem.

Theorem 1 (Baye). *Let A and B be events. Then the conditional probability of A given B is calculated by the following*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

The backward probability is $P(C_{g-1}|C_g)$, and by Baye's theorem we get

$$P(C_{g-1}|C_g) = \frac{P(C_g|C_{g-1})P(C_{g-1})}{P(C_g)}$$

$P(C_g|C_{g-1})$ is somewhat easy to calculate, it is either P_C (1) or P_M (2). $P(C_g)$ and $P(C_{g-1})$ is the probability of observing the configurations C_g and C_{g-1} . They are

both calculated by the recursion. We thereby make use of the exact probabilities for the subsequences. To save some time, we recognize that when calculating $P(C_g)$, $P(C_{g-1})$ is automatically calculated.

For each allele in each subsequence we can calculate u_{km} corresponding to Hobolth's u_{km} .

$$u_{km}(\theta) = \begin{cases} \frac{P(C_g|C_{g-1,k})P(C_{g-1,k})}{P(C_g)} \cdot \frac{N_{m,k}}{n_{m,k}} & \text{if } n_{m,k} \geq 2 \text{ or } k \in \mathbb{M}_m \\ 0 & \text{if } n_{m,k} = 1 \text{ and } k \notin \mathbb{M}_m \end{cases}$$

The new proposal:

$$Q_{\theta_0}^{new}(k|\mathbf{S}, \mathbf{N}) \propto \begin{cases} \sum_m u_{km}(\theta_0) & \text{if } N_k \geq 2 \text{ or } k \in \mathbb{M} \\ 0 & \text{if } N_k = 1 \text{ and } k \notin \mathbb{M} \end{cases}$$

9 Results

To compare the new proposal to the Hobolth and the Stephens-Donnelly proposal, we have considered the following two data-sets.

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{N} = \begin{pmatrix} 2 \\ 2 \\ 4 \end{pmatrix}$$

Dataset A.

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{N} = \begin{pmatrix} 4 \\ 1 \\ 2 \\ 1 \\ 3 \\ 3 \end{pmatrix}$$

Dataset B.

9.1 Calculations based on data-set A

Consider the data-set A. We have calculated the likelihood surfaces, the standard errors for the three proposals and the ESS using the three different proposals. We used 10,000 samples and the driving value $\theta = 2$. The calculated *ESS* and an estimate for the *ESS* is shown in the table below.

Proposal	ESS
Stephens-Donnelly	4,920.4
Hobolth	6,177.6
New (1 site)	6,143.7
New (2 sites)	6,444.2
New (3 sites)	7,280.7

As we would expect the $ESS_{SD} < ESS_{Hobolth}$ and $ESS_{Hobolth} \approx ESS_{New1}$. Also the ESS gets higher, as we consider more sites in each subsequence in the new proposal. In figure 8 to figure 12 on the following page is shown the estimated likelihood surfaces.

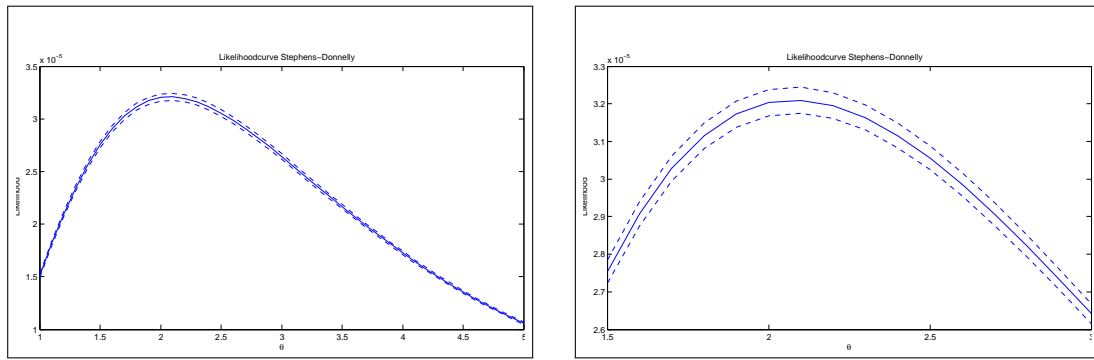


Figure 8: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset A. The proposal distribution is the Stephens-Donnelly proposal.

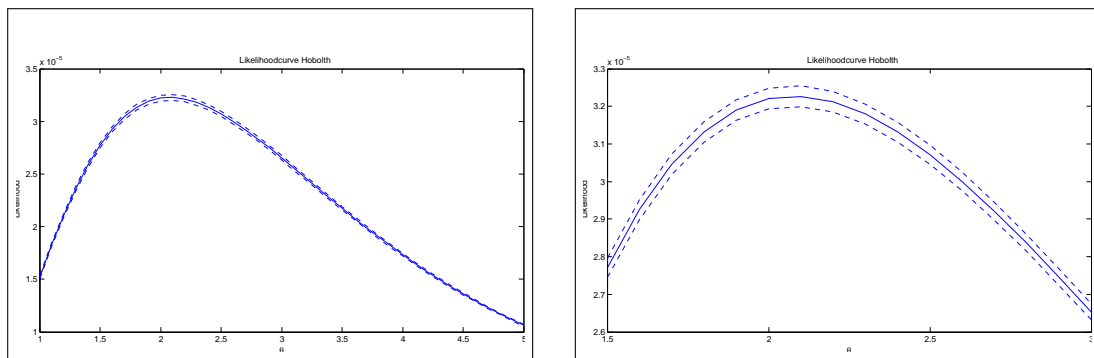


Figure 9: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset A. The proposal distribution is the Hobolth proposal.

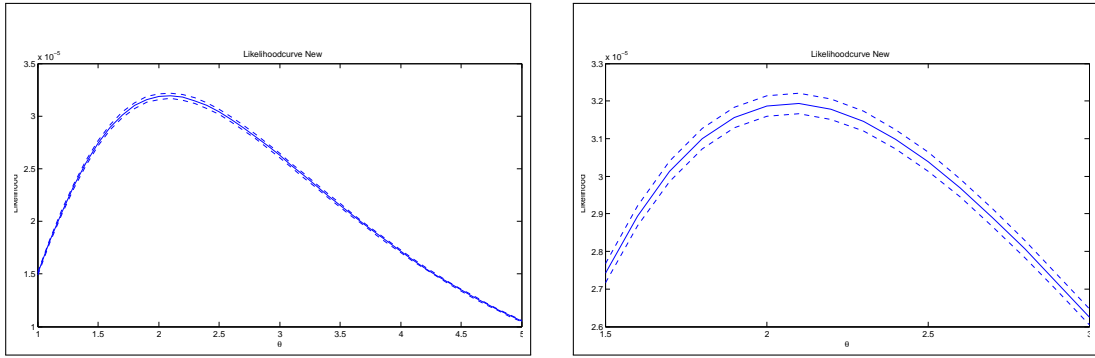


Figure 10: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset A. The proposal distribution is the New proposal considering 1 site in each subsequence.

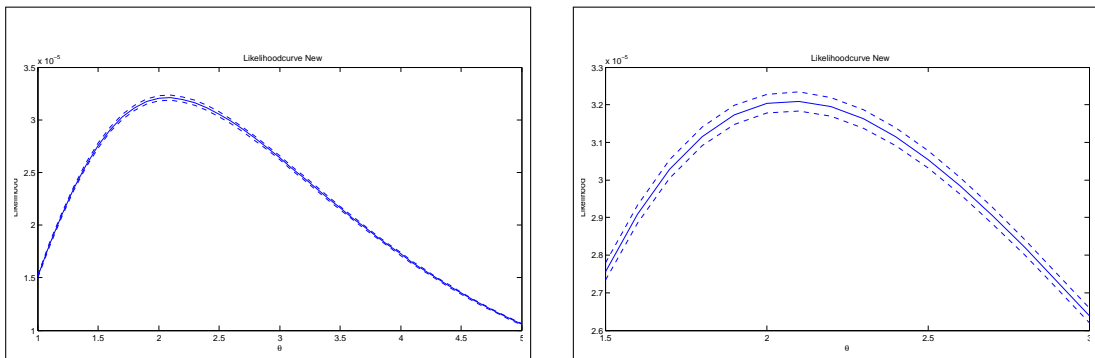


Figure 11: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset A. The proposal distribution is the New proposal considering 2 sites in each subsequence.

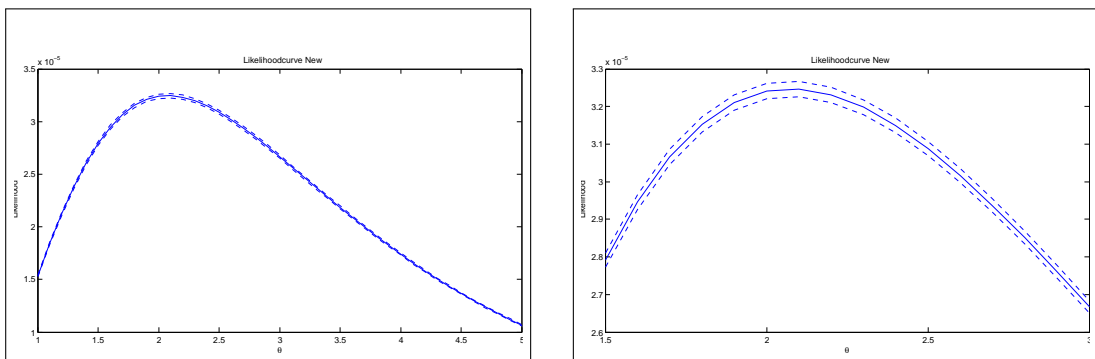


Figure 12: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset A. The proposal distribution is the New proposal considering 3 sites in each subsequence.

9.2 Calculations based on data-set B

Consider now the data-set B. As before we have calculated the likelihood surfaces, the standard errors for the three proposals and the ESS using the three different proposals. We used 10,000 samples and the driving value $\theta = 4$. The calculated ESS and an estimate of the ESS is shown in the table below.

Proposal	ESS
Stephens-Donnelly	726.0
Hobolth	2,150.8
New (3 sites)	3,654.8
New (4 sites)	4,316.8

As we would expect the $ESS_{SD} < ESS_{Hobolth}$ and $ESS_{Hobolth} < ESS_{New3}$. Also the ESS gets higher, as we consider more sites in each subsequence in the new proposal. In figure 13 to figure 16 on the following page is shown the estimated likelihood surfaces.

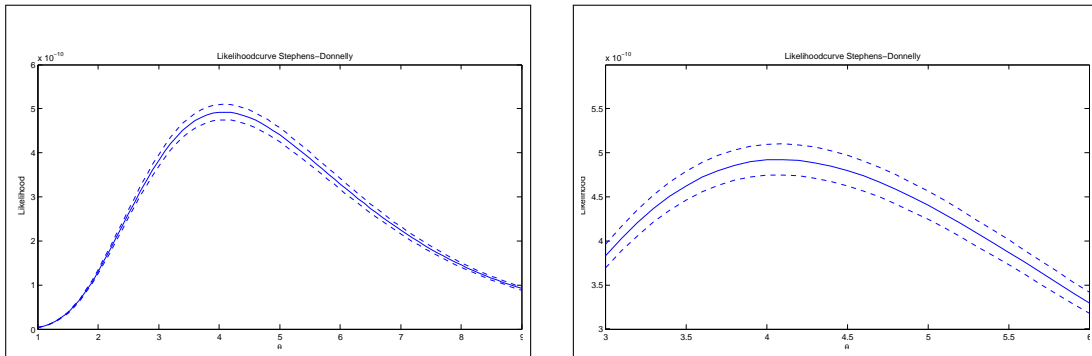


Figure 13: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset B. The proposal distribution is the Stephens-Donnelly proposal.

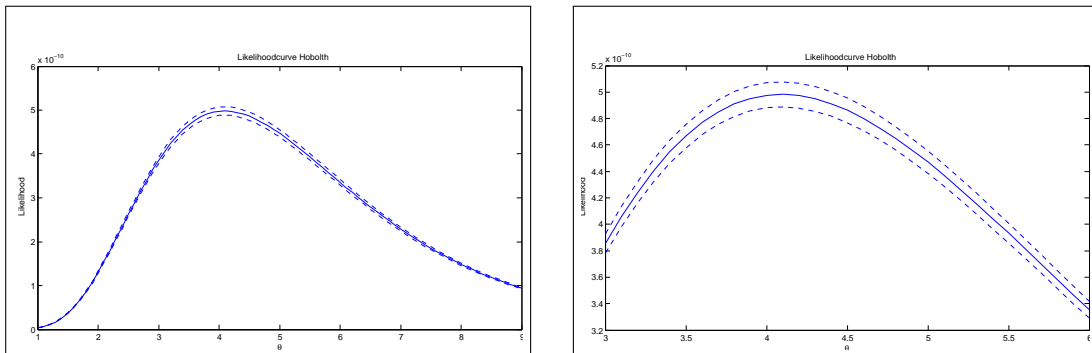


Figure 14: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset B. The proposal distribution is the Hobolth proposal.

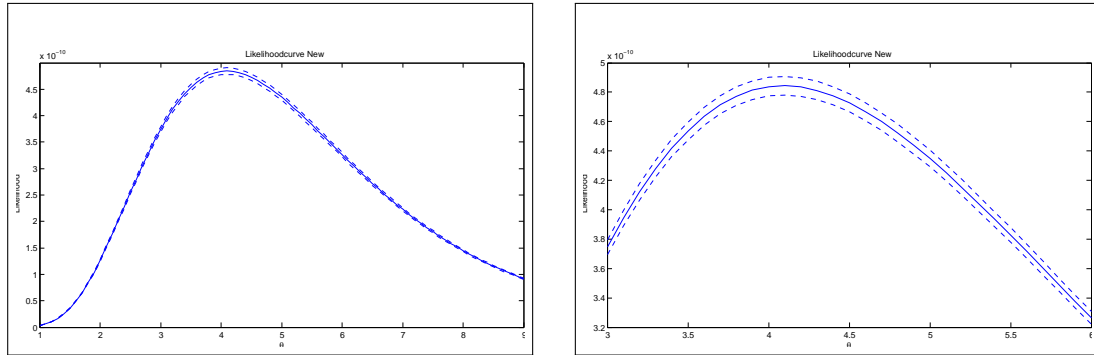


Figure 15: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset B. The proposal distribution is the New proposal considering 3 sites in each subsequence.

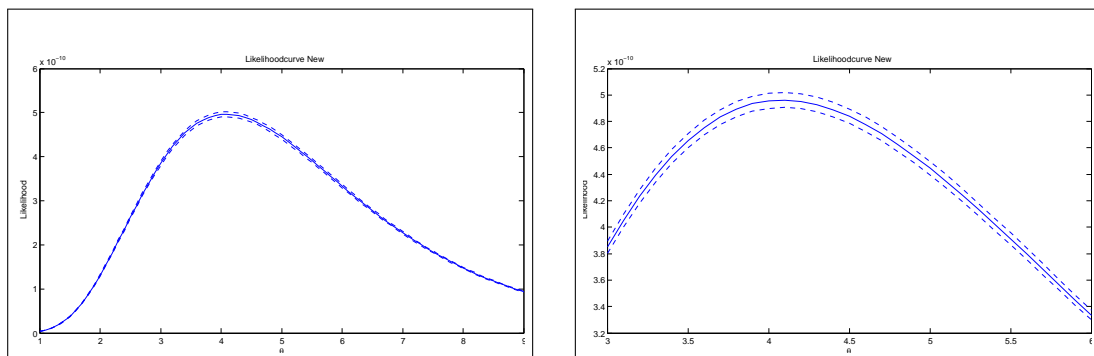


Figure 16: Estimated likelihood surfaces (full line) and standard errors (dashed line) for the dataset B. The proposal distribution is the New proposal considering 4 sites in each subsequence.

10 Discussion and conclusion

Discussion - How to improve our proposal

We have to consider a problem which arises when we apply the recursion to subsequences. We cannot be sure that local events in a subsequence are compatible with the global order of the whole data-set. Looking deeper into this problem we realise that we have to take the context into concern.

We have to convince ourselves how an event for a subsequence is dependent on the rest of the data. To remove a mutation in a subsequence is independent from the rest of the data-set. This event only depends on the site, and a subsequence covers the entire site. Coalescence however has to be dealt with more carefully. We cannot

just coalesce two alleles that are identical in a subsequence; they might differ in another subsequence, making it impossible for them to coalesce. To start with, this problem can be dealt with quite easily. Realizing that the multiplicity for the whole data-set fits to any given subsequence of the data-set, we can distinguish between identical alleles in the subsequence. This is illustrated below

$$\mathbf{S}_{\text{sub}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{N} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \quad (11)$$

In (11) we have cut out \mathbf{S}_{sub} as a subsequence with three sites. \mathbf{N} is the multiplicity of the whole data-set and of \mathbf{S}_{sub} . If we observed \mathbf{S}_{sub} , as a data-set, we would simply merge row 1 and 2 because they are identical. But because they are separated in \mathbf{N} , they must differ somewhere in the data-set by one or more mutations.

This tells us, that row 1 and 2 should not merge at least until one of them has reduced its multiplicity to 1. The problem is to decide when to merge row 1 and 2, and whether row 3 after losing its mutation should merge with 1, 2 or none of them. We have to know how to merge rows, because we calculate the probability of observing each individual subsequence with the recursion (5). We modified our representation of the subsequences in a rather naive way, to adjust it to our implementation of the recursion.

We will discuss two methods to decide when to merge two identical rows in a subsequence. The first method is a rough improvement still far from an optimal restriction. The quality of the first method is that it should be fairly easy to implement because it is a general restriction. Second method is more specific for the actual data-set, and therefore somewhat more complex. Second method takes the global state into account, considering the mutations that make two rows as 1 and 2 in (11) differ.

First method

If a row with multiplicity one is identical to any other row in the subsequence, let it merge with the one of the identical rows with the highest multiplicity.

This could happen if either 2 coalesced or 3 had its mutation removed in (11). Both would result in merging with row 1 because it has the highest multiplicity. This is justified by looking at P_M (2) which states that a mutation is most likely to happen on the allele with the highest multiplicity. As row 2 and 3 only can differ from row 1 by mutations elsewhere in the data-set, we suggest that they originated from row 1, and then mutated at some point.

Second method

The second method is based on the idea, that we can take the global information into account. We would like to store the information about the rest of the data-set when looking at a subsequence of the data-set.

This method would tell us exactly where row 1 and 2 are different from each other. We could then use this information to tell when to merge the two rows. If at some place on the global scale, row 2 has a mutation, which is not present on row 1, this must be removed before we can merge the two rows. One way to do this is to redefine the probability of row 2 when its multiplicity is one. The probability for row 2 to merge with row 1 is then the probability of removing a mutation from row 2.

Conclusion

In this report we have introduced a new importance sampler for the ISM. Our importance sampler takes ancestral states and the local information into account. We have shown that our new proposal beats both SD and Hobolth proposals as for the ESS values. The time race though we have not won. SD and Hobolth proposals are significantly faster to do their calculations than our proposal. This could probably be improved quite a lot if our implementation were rewritten. We have discussed how to further improve the importance sampler, focusing on the global information. We would expect to get even better results with the restrictions implemented in the code.

11 Acknowledgements

We would like to thank Jotun Hein, Rune Lyngsø and Paul Jenkins for their valuable guidance and discussions throughout this project. A special thanks goes to Ferenc Huzár for helpful programming assistance. We were both supported by University of Århus for which we are very grateful. We would also like to thank Carsten Wiuf, Istvan Miklos and the rest of the summerschool students at SPR2. They were all very helpful and gave us a wonderful and educative summer.

Literature

HEIN, J., SCHIERUP, M.H. and WIUF, C. (2006). *Gene Genealogies, Variation and Evolution*. Oxford University Press Inc. New York.

HOBOLTH, A., UYENOYAMA, M. and WIUF, C. (2008). *Importance Sampling for the Infinite Sites Model*

LIU, J.S. (2001). *Monte Carlo strategies in scientific computing*. Springer-Verlag New York.