Rarecoal

Stephan Schiffels

The rarecoal coalescent framework

This model describes a coalescent framework for rare alleles. We define rare alleles roughly by requiring i) the allele count of the derived mutation to be small, typically not larger than 10, and ii) the total number of samples to be much larger, say 100 or more. The idea is to provide a general approach of computing the joint allele frequency spectrum for rare alleles under an arbitrary demographic model under population splits and population size changes. Migration and admixture will be incorporated in the future.

Definitions

In the following, we compute the probability to observe a pattern of rare alleles seen across multiple populations, given a demographic model. In the simplest case, a demographic model is tree-like and consists of population split times and constant population sizes in each branch of the tree. Time is counted backwards in time, with $t = 0$ denoting the present and $t > 0$ denoting scaled time in the past. We denote the scaled coalescence rate (scaled inverse population size) in population k at time t by $\lambda_k(t) = N_0/N_k(t)$, where $N_k(t)$ is the population size in population k at time t, and N_0 is a scaling constant which we set to $N_0 = 20000$ for modeling human evolution.

We consider a number of P subpopulations. We define a vector $\mathbf{n} = \{n_k\}$ for $k = 1 \ldots P$ summarizing the number of sampled haplotypes in each population. We also define vector $\mathbf{m} = \{m_k\}$ as the set of derived allele counts at a single site in each population. As an example, consider 5 populations with 200 haplotypes sampled in each population, and a rare allele with total allele count 3, with one derived allele seen in population 2 and 2 derived alleles seen in population 3. Then we have $\mathbf{n} = \{200, 200, 200, 200, 200\}$ and $\mathbf{m} = \{0, 1, 2, 0, 0\}$.

Looking back in time, lineages coalesce and migrate, so the numbers of ancestral and derived alleles in the past decrease over time. In theory one needs to consider a very large state space of configurations for this process, with one state for each possible number of ancestral and derived lineages in each population. Here we make a major simplification: While we will consider the full probability distribution over the derived lineages, we will consider only the expected number of ancestral alleles over time. Specifically, we define the expected number of ancestral alleles in population k at time t as $\mathbf{a}(t) = \{a_k(t)\}$. For the derived alleles, we define a state $\mathbf{x} = \{x_k\}$ as a configuration of derived lineages in each population. The probability for state $\mathbf{x}$ at time t is defined by $b(\mathbf{x}, t)$.

Coalescence

We now consider the evolution of the two variables $a(t)$ and $b(\mathbf{x}, t)$ through time under the standard coalescent. We first introduce a time discretization. We define time points $t_0 = 0, \ldots t_T$. Here, $t_T = t_{\max}$ should be far enough in the past to make sure that most lineages have coalesced by then with a high probability. We choose a time patterning that is linear in the beginning and crosses over to an exponentially increasing interval width. Specifically, the patterning follows this equation, inspired by the time discretization in [1]:

$$t_i = \alpha \exp\left(\frac{i}{T} \log\left(1 + \frac{t_{\max}}{\alpha})\right)\right) - \alpha. \tag{1}$$

Here, T is the number of time intervals, and α is a parameter that controls the crossover from linear to exponential scale. In practice, we use $\alpha = 0.01$, $t_{\max} = 20$ and $T = 3044$, which are chosen such that the initial step width equals one generation (in scaled units with $N_0 = 20000$), and the crossover scale is 400 generations.

Given the number of sampled haplotypes in each population n_k, and the observed number of derived alleles m_k in each population, we initialize our variables as follows:

$$a_k(t=0) = n_k - m_k. \tag{2}$$

for each population k, and

$$b(\mathbf{x}, t=0) = 1 \text{ if } x_k = m_k \text{ for all } k = 1 \dots P \tag{3}$$

$$b(\mathbf{x}, t=0) = 0 \text{ otherwise} \tag{4}$$

Under a linear approximation, we can compute the value of $\mathbf{a}$ at a time point $t + \Delta t$, given the value at time t:

$$a_k(t + \Delta t) = a_k(t) \left(1 - \frac{1}{2}(a_k(t) - 1)\lambda_k(t)\Delta t \right). \tag{5}$$

The factor 1/2 corrects overcounting: any one coalescence takes one of two lineages out, so it should be counted half per participating lineage. We can improve this update equation slightly beyond the linear approximation: In the limit of $\Delta t \to 0$, equation 5 forms a differential equation which can be solved for finite intervals Δt:

$$a_k(t + \Delta t) = \frac{1}{1 + \left(\frac{1}{a_k(t+\Delta t) - 1} \right) \exp\left(-\frac{1}{2}\lambda_k(t + \Delta t) \times (t + \Delta t) \right)}. \tag{6}$$

For the derived alleles, we need to update the full probability distribution $b(\mathbf{x}, t)$:

$$\begin{aligned} b(\mathbf{x}, t + \Delta t) =& b(\mathbf{x}, t) \exp\left(-\sum_k \left(\binom{x_k}{2} \lambda_k(t) + x_k a_k(t) \lambda_k(t) \right) \Delta t \right) \\ &+ \sum_l b(x_1 \dots (x_l + 1) \dots x_P, t) \left(1 - \exp\left(\binom{x_l + 1}{2} \lambda_l(t) \Delta t \right) \right) \end{aligned} \tag{7}$$

where the first term accounts for the reduction of the probability over time due to derived lineages coalescing among themselves or coalescing with an ancestral lineage, and the second term accounts for the increase from those two processes occurring in states with a higher number of derived lineages. In contrast to the equation for $a(t)$, we cannot solve this as a differential equation and will only use this linear approximation in Δt.

Population Splits

Population splits forward in time are joins backward in time. We consider a population join backward in time from population l into population k. For the ancestral lineages, this means that after the join, population k contains the sum of lineages from population k and l:

$$a'_k(t) = a_k(t) + a_l(t) \tag{8}$$

$$a'_l(t) = 0 \tag{9}$$

where the primed variable marks the variable after the event, which will then be used as the basis for the next coalescence update.

For the derived lineages, we need to sum probabilities in the correct way. We first define a transition function that changes a state before the join to new states after the join:

$$\mathbf{x}' = J(\mathbf{x}), \tag{10}$$

where

$$J((\dots x_k \dots x_l \dots)) = (\dots (x_k + x_l) \dots 0 \dots) \tag{11}$$

We can then define the join itself as a sum over all states before the join that give rise to the same state after the join:

$$b'(\mathbf{x}', t) = \sum_{\mathbf{x}, J(\mathbf{x}) = \mathbf{x}'} b(\mathbf{x}, t) \tag{12}$$

The likelihood of a configuration of rare alleles

Eventually we want to compute the probability for a given configuration ($\mathbf{n}$, $\mathbf{m}$) observed in the present. This probability is equal to the probability that a) all derived lineages coalesce before any of them coalesces to any ancestral-allele lineage, and b) that a mutation occurred on the single lineage ancestral to all derived lineages.

We define a singleton state $\mathbf{s}^k$ to be the state in which only $x_k = 1$ and $x_l = 0$ for $l \neq k$. We accumulate the total probability for a single derived lineage:

$$d(t + \Delta t) = d(t) + \sum_k b(\mathbf{s}^k)\Delta t. \tag{13}$$

Then the likelihood of the configuration under the model is

$$L(\mathbf{n}, \mathbf{m}) = \mu d(t_{\max}) \prod_{k=1}^{P} \binom{n_k}{m_k}, \tag{14}$$

which is the total probability of a mutation occurring on a single derived lineage, times the number of ways that $\mathbf{m}$ derived alleles can be drawn from a pool of $\mathbf{n}$ samples. Note that $d(t_{\max})$ depends on $\mathbf{n}$, $\mathbf{m}$ and the demographic parameters, which we have omitted for brevity so far.

Parameter estimation

The above framework presents a way to efficiently compute the probability of observing a distribution of rare alleles, $\mathbf{m}$ for a large number of samples $\mathbf{n}$ in multiple subpopulations, given a demographic model. We can summarize the full data as a histogram of rare allele configurations. We denote the ith allele configuration by $\mathbf{m}_i$ and the number of times that this configuration is seen in the data by $N(\mathbf{m}_i)$. We then write

$$\mathcal{L}(\{N(\mathbf{m}_i)\}|\Theta) = \prod_i L(\mathbf{m}_i|\Theta)^{N(\mathbf{m}_i)}, \tag{15}$$

where we have introduced a meta-parameter Θ that summarizes the entire model specification (population split times and branch population sizes), and we have made the dependency of L (eq. 14) on Θ explicit. For brevity we have omitted the sample sizes $\mathbf{n}$. For numerical purpose, we always consider the logarithm of this:

$$\log \mathcal{L}(\{N(\mathbf{m}_i)\}|\Theta) = \sum_i N(\mathbf{m}_i) \log L(\mathbf{m}_i|\Theta). \tag{16}$$

The sum in equation 16 comprises all possible configurations in the genome, in principle. In practice, we only explicitly compute it for configurations between allele count 1 and 4, and replace the rest of the counts with a bulk probability:

$$\log \mathcal{L}(\{N(\mathbf{m}_i)\}|\Theta) = \sum_i I(\mathrm{AC}(i)) N(\mathbf{m}_i) \log L(\mathbf{m}_i|\Theta) + N_{\text{other}} \log L_{\text{other}}(\Theta), \tag{17}$$

where the indicator function $I(\mathrm{AC}(i))$ gives 0 if the allele count is between 1 and 4, and 0 otherwise. The bulk count N_{other} simply counts up sites with either no variant or variants with allele count larger than 4. The bulk probability is simply:

$$L_{\text{other}}(\Theta) = 1 - \sum_i (1 - I(\mathrm{AC}(i)) L(\mathbf{m}_i|\Theta), \tag{18}$$

With a given population tree and a given histogram of allele configuration counts $N(\mathbf{m}_i)$, we implemented numerical optimizations over the parameters Θ to find the maximum likelihood parameters, and MCMC to estimate the posterior distributions for all parameters given the data. We usually first search for the maximum with the optimization method, which is much faster than MCMC, and then use MCMC to explore the distribution around that maximum.

Implementation

We implemented this method in the Haskell programming language as a program called "rarecoal", available from github at `https://github.com/stschiff/rarecoal.hs`.

Testing Rarecoal with simulated data

We defined a simple population-tree, as shown in Figure 3b of the paper. We used the SCRM simulator [2] with the following command line to simulate 20 chromosomes of 100Mb:

```
   scrm 1000 1 -l 100000 -t 100000 -r 80000 100000000 -I 5 200 200 200 200 200 -ej 0.00125
2 1 -ej 0.0025 4 3 -ej 0.00375 5 3 -ej 0.005 3 1 -en 0.00000001 1 0.1 -en 0.00000002 2 2.0
-en 0.00000003 3 1.0 -en 0.00000004 4 5.0 -en 0.00000005 5 10.0 -en 0.00125001 1 1.0 -en
0.0025001 3 0.5 -en 0.00375001 3 0.8 -en 0.005001 1 1.0.
```

The tree topology of this tree is `(((0, 1), ((2, 3)), 4))`, with branches ordered left to right as in Figure 3b. We first obtained maximum likelihood estimates of only the split times, and a globally fixed population size. Note: all times are scaled with $2N_0$ (not $4N_0$ as in the command line above), and all population sizes are scaled by N_0. This first round of maximization is summarized in the following table:

Parameter	**True value**	**Initial value**	**Estimate**
$t_{(0,1)}$	0.0025	0.001	0.00271
$t_{(2,3)}$	0.005	0.002	0.00242
$t_{((2,3),4)}$	0.0075	0.003	0.00452
$t_{(((0,1),((2,3)),4))}$	0.01	0.004	0.00592
N_{global}	1	1	0.859

We then used these estimates as starting point for the full model optimization, with separate population size estimates in each internal and leaf-branch of the tree. We denote the population size parameters with N, using as subscript the subtree of the node below that branch. The results are summarized in the following table, including confidence intervals for each parameter as obtained by MCMC:

Parameter	**True Value**	**Median Estimate**	**95% CI**
$t_{(0,1)}$	0.0025	0.00266	(0.00265, 0.00268)
$t_{(2,3)}$	0.005	0.00497	(0.00495, 0.00499)
$t_{((2,3),4)}$	0.0075	0.00814	(0.00812, 0.00816)
$t_{(((0,1),((2,3)),4))}$	0.01	0.00965	(0.00963, 0.00967)
N_0	0.1	0.1013	(0.1012, 0.1014)
N_1	2	2.30	(2.28, 2.31)
N_2	1	0.995	(0.992, 0.998)
N_3	5	4.98	(4.95, 5.01)
N_4	10	10.54	(10.51, 10.58)
$N_{(0,1)}$	1	0.9315	(0.9309, 0.9322)
$N_{(2,3)}$	0.5	0.6123	(0.6122, 0.6130)
$N_{((2,3),4)}$	0.8	0.4648	(0.4647, 0.465)
$N_{(((0,1),((2,3)),4))}$	1	0.928	(0.92, 0.934)

In most parts of the tree, the estimates are close to the truth, with one exception: the worst fit parameter is $N_{((2,3),4)}$, the ancestral population size in the branch preceeding the second split, which is about 40% too low. This may be due to the fact that this branch is relatively short and the subtree below has relatively large population sizes, which are both causes of relatively low amounts of genetic drift and consequently relatively weak information in the data about parameters.

Learning the European population tree

We started with three populations (FIN, IBS, NED) and tested all three possible tree topologies for these populations, with one global population size. The best tree is ((FIN, NED), IBS) with scaled split times 0.0039 and 0.006, and a global populaton isze of 2.3.

We then added the Danish branch and tested every possible point in the tree to join. The maximum likelihood point to join was the Dutch branch at time 0.0028, resulting in the topology ((FIN, (NED, DMK)), IBS). We then maximized split times and global population size on that tree and found split times 0.003, 0.0038 and 0.006 with a global population size of 2.34.

Next, we added the TSI as additional population to the tree and first again checked every possible point in the tree to merge. We found that the maximum likelihood point in the tree was - surprisingly - on the Danish branch at an extremely recent time 0.0001. We decided this to be some artifact of the fixed population sizes and chose the second-highest merge-point, which was the Spanish branch at time 0.0023, resultin a topology ((FIN, (NED, DMK)), (IBS, TSI)). Using this merge-point and the previous parameters as initial parameters, we then again estimated maximum likelihood parameters for this five-population tree and found parameters summarized in the following table:

Parameter	**Estimate**
$t_{\texttt{(NED,DMK)}}$	0.0024
$t_{\texttt{(FIN,(NED, DMK))}}$	0.0032
$t_{\texttt{(IBS,TSI)}}$	0.0049
$t_{\texttt{((FIN,(NED,DMK)),(IBS,TSI))}}$	0.0062
N_{global}	3.15

We then allowed for separate population sizes within each branch of the tree and inferred parameters using maximization and subsequent MCMC. The results are as follows:

Parameter	**Estimate**
$t_{\texttt{(NED,DMK)}}$	0.0039
$t_{\texttt{(FIN,(NED, DMK))}}$	0.004
$t_{\texttt{(IBS,TSI)}}$	0.0054
$t_{\texttt{((FIN,(NED,DMK)),(IBS,TSI))}}$	0.0064
N_{FIN}	0.53
N_{IBS}	8.23
N_{TSI}	6.89
N_{NED}	8.37
N_{DMK}	1.87
$N_{\texttt{(NED,DMK)}}$	1.05
$N_{\texttt{(FIN,(NED, DMK))}}$	0.94
$N_{\texttt{(IBS,TSI)}}$	983.25
$N_{\texttt{((FIN,(NED,DMK)),(IBS,TSI))}}$	2.00

Finally, we added the British population branch, by first again trying every possible point for it to merge into the tree. We found that the most likely point to merge was on the Netherland branch at time 0.0007. We used this as a starting point for another round of parameter estimation, and found that the resulting tree had two suspiciously close population splits, with an essesntially star-like split of GBR, NED and FIN. We therefore changed the topology and tried whether merging the GBR population into the Finnish branch would give a higher likelihood. Indeed this was the case, so the best fitting tree topology is (((FIN, GBR), (NED, DMK)), (TSI, IBS)). The final parameter estimates are:

Parameter	**Median Estimate**	**95% CI**
$t_{\texttt{(NED,DMK)}}$	0.00372	(0.00370, 0.00373)
$t_{\texttt{(FIN, GBR)}}$	0.00399	(0.00398, 0.004)
$t_{\texttt{((FIN, GBR),(NED, DMK))}}$	0.00417	(0.00415, 0.00418)
$t_{\texttt{(IBS,TSI)}}$	0.00238	(0.00237, 0.00240)
$t_{\texttt{((FIN, GBR),(NED,DMK)),(IBS,TSI))}}$	0.00605	(0.00603, 0.00607)
N_{FIN}	0.54868	(0.54863, 0.54874)
N_{GBR}	4.353	(4.347, 4.358)
N_{IBS}	4.910	(4.908, 4.913)
N_{TSI}	4.3263	(4.3253, 4.3272)
N_{NED}	10.500	(10.488, 10.508)
N_{DMK}	1.755	(1.741, 1.771)
$N_{\texttt{(NED,DMK)}}$	1.060	(1.059, 1.061)
$N_{\texttt{(FIN, GBR)}}$	0.85	(0.85, 0.85)
$N_{\texttt{((FIN, GBR),(NED, DMK))}}$	0.86	(0.86, 0.86)
$N_{\texttt{(IBS,TSI)}}$	998	(992, 1000)
$N_{\texttt{((FIN, GBR),(NED,DMK)),(IBS,TSI))}}$	1.8149	(1.8137, 1.8169)

We also tried whether the high ancestral population size of the IBS/TSI branch was a sub-optimal local maximum, by restarting the MCMC from a lower population size and an earlier IBS/TSI split time. This resulted in similar estimates as the ones presented above, so we conclude that this tree is the maximum likelihood tree.

Mapping individuals onto the tree

For mapping the ancient individuals onto the tree we first generate data sets consisting of all the European individuals that went into learning the European tree, plus one additional individual. We then compute the likelihood for a family of models, which are all composed of the original model learned for the European populations, plus one more population that merges onto the tree. We vary only the merge point of that additional population, over all leaf- and internal branches of the Europena tree, with a discretized time interval of scaled time 0.0001. In this likelihood computation, we deviated from the standard likelihood (equation 17) in one detail: We only explicitly fitted sites in which the ancient samples carried the derived allele. All other sites were accumulated into the bulk number (N_{other}) alongside variants with allele count higher than 4 and sites without variants.

We tested this approach with individuals from the 1000 Genomes project [3], which for this analysis were taken out of the reference set of FIN, GBR, IBS and TSI samples. As seen in Extended Data Figure 8, all of the 8 individuals shown map onto the tip of their population branch, as exptected for samples that belong to those reference populations. For the GBR individuals, we also noticed some systematic deviations, as shown in Extended Data Figure 8, with some samples mapping to the common ancestor of all Northern European populations. We believe this is due to population structure within the GBR samples in 1000 Genomes, which were sampled from three locations (Kent, Cornwall and Orkney). Because our European tree assumes one panmictic population for those subpopulations we expect some samples to not be represented by this population. Note that we do not know the sampling locations of any individual in 1000 Genomes, so cannot separate them on the European tree in the first place.

References

[1] Li, H., and Durbin, R. (2011). Inference of human population history from individual whole-genome sequences. http://doi.org/10.1038/nature10231

[2] Staab, P. R., Zhu, S., Metzler, D., and Lunter, G. (2015). scrm: efficiently simulating long sequences using the approximated coalescent with recombination. Bioinformatics, btu861. http://doi.org/10.1093/bioinformatics/btu861

[3] 1000 Genomes Project. (2015). A global reference for human genetic variation. Nature (in Revision).