# CH3: Coalescent Theory and Ancestral Recombination Graphs

CS 182 Spring 2024

*Scribes (from previous years)*: jwang194, spurkaya, dbenisvy

*Scribes (2022)*: jchang57, ihuang8, vkasibha, bkirz, ilee26, nlee16, yliang6, mmai1, spyon, jscheric, zzhu42

*Compiled & edited by* eyouth, jwang194

*Reach out to cs1820tas@lists.brown.edu for any clarifications or corrections.*

## Motivation

We are often curious about evolutionary history; by it, we can understand selective pressures and conserved function. For example, with a complete evolutionary history, we might be able to identify the process by which a pathogen shifted hosts or propose hypotheses for heavily but erratically conserved genes. Ideally, this history would be complete, with a record of each living thing, its genotype, descendants, and the time between generations. But of course, there is no way for us to keep a uncover all of this information. We make do with simple models, such as phylogenetic trees, which track only genotypes and model evolution as a simple mutational process.

We are aware, however, of another fundamental "form" of evolution: recombination. In many ways, this is "the" primary source of major genetic variation in eukaryotes, but it also introduces a problem: even when we consider only haplotypes, a genomic sequence can have *two* related parents, producing an (undirected) cycle in the ancestry graph. The class of such graphs (DAGs) is much more complex than the class of all trees, and reconstructing evolutionary histories is correspondingly more difficult.

As an aside, we should contrast this with a more comparative genomics approach. There, we are given a pair of sequences between which we seek significant similarity, for the sake of identifying and understanding regions of biological interest. Here, in coalescent theory, we begin with a set of genotypes which we assume to be related and wish to understand the evolutionary process which produced them (a new problem!).

## Overview of ARGs

Ancestral recombination graphs (ARGs) are constructed from haplotypes (note, this means that genotypes must be *phased*) in a population to infer historical patterns of mutation and recombination which resulted from diversification of a common ancestor. Coalescent theory is a population genetics model which assumes that variant alleles within a population arose over time through random mutation of ancestral alleles, thereby enabling reconstruction of inheritance patterns stretching back many generations. Since the process of inferring a population's precise genetic lineage is complex, many algorithms have been devised to accurately reconstruct ARGs from a given set of haplotypes, using principles of Mendelian inheritance and maximum parsimony. Such ARGs can yield insights into patterns of trait inheritance and even suggest causative alleles for diseases shared between individuals.

# Reconstruction Theory

Consider a population of individuals belonging to two classes: “unaffected” and “affected” (or “healthy” and “diseased”, if applied to a specific disease trait). Genotypic information from these individuals can be used to reconstruct a geneological tree (an ARG) for the population which offers a putative evolutionary history. Each genotype is the confluence of two haplotypes, which are themselves sequences of single nucleotide polymorphisms (SNPs). Each SNP is assumed to be biallelic, and the two bases present at a given SNP are arbitrarily assigned the labels 0 and 1. Thus, haplotypes are ordered binary strings, while genotypes are ordered ternary strings (with the value at a given SNP equal to that of both haplotypes if identical at that SNP, and denoted with a 2 if the haplotypes differ at that SNP).

The challenge of inferring parental haplotypes given ambiguous genotypes is known as the *phasing problem*, and sophisticated algorithms have been developed to “phase” genotypes in a population in a manner which maximizes repeated haplotypes. However, the phasing problem is beyond the scope of this chapter, and the ARGs considered here will be derived from haploid data only.

In such an ARG, individuals are represented as leaves, ancestral genotypes are represented by nodes, and “events” are represented by edges. “Events” are defined as follows:

**Defn**: A *coalescence* is a merger of two haplotypes which at every SNP either agree (i.e., both are 0 or both are 1) or are partially unresolved (i.e., one SNP is denoted with a $\bullet$ symbol, indicating that its identity is not defined)

**Defn**: A *mutation* is a change in a single SNP (i.e., from 0 to 1 or from 1 to 0)

**Defn**: A *recombination* is a merger of two haplotypes at a specific SNP position, such that the “recombinant” haplotype shares identity before the breakpoint with one “parental” sequence and after the breakpoint with the other sequence

Coalescence events provide a means of “condensing” the pool of haplotypes following inference of recombination and mutation events, ultimately enabling full resolution of the population to the single common ancestor at the root of the ARG. Mutation events may suggest putative origins for specific traits; if many or all of the “descendents” of the ancestral mutant node carry the trait, it may suggest a causative role for the allele of interest. Recombination events can be conceptualized as “crossing over” of parental branches; all “descendents” of recombinant nodes will share the same recombinant signature (barring further mutations). *Marginal trees* can be extracted from the regions between recombination breakpoints.

# The Minichiello-Durbin Algorithm

In 2006, Mark Minichiello and Richard Durbin introduced an algorithm for inferring geneologies from population genotype data. Their algorithm enabled fine disease mapping and interpretation of results from other large-scale association studies (e.g., GWAS).

**Input**: A population of genotypes (or haplotypes), with individuals labelled for a given trait

**Output**: An ARG which explains inheritance patterns of the trait across the population

Since the “true” ARG is generally unknown, the Minichiello-Durbin algorithm employs heuristics to infer many possible ARGs, and conducts statistical tests to determine the “most likely” patterns of inheritance which explain the occurrence of the trait in question. Each ARG is constructed “backward in time”, with each step corresponding to one of the events defined above (coalescence, mutation or recombination).

## Notation

Let $S_T$ be the set of haplotypes present at time $T$, such that each haplotype is an $m$-mer drawn from the alphabet $\{0, 1, \bullet\}$ (where $\bullet$ denotes an “undefined” allelic state or an unrepresented ancestral allele). Let $C[i]$ denote the allelic state of haplotype $C$ at SNP $i$ for all $1 \leq i \leq m$, and define the following relations (similarity and complement):

$$C_1[i] \sim C_2[i] \text{ iff } C_1[i] = C_2[i] \text{ or } C_1[i] = \bullet \text{ or } C_2[i] = \bullet$$

$$\neg\, C[i] = \begin{cases} 1 & \text{if } C[i] = 0 \\ 0 & \text{if } C[i] = 1 \\ \bullet & \text{if } C[i] = \bullet \end{cases}$$

**Defn**: A *shared tract* is an identical subsequence shared by two haplotypes, denoted $[C_1, C_2][a, b]$, if the following conditions (similarity, definition and maximal range) are satisfied:

1. $C_1[i] \sim C_2[i]$ for all $a \leq i \leq b$
2. $C_1[k] \neq \bullet$ or $C_2[k] \neq \bullet$ for some $a \leq k \leq b$
3. $(C_1[a-1] \nsim C_2[a-1]$ if $a > 1)$ and $(C_1[b+1] \nsim C_2[b+1]$ if $b < m)$

Let $T = 1$ denote the "contemporary" time, so that $S_1$ represents the initial set of haplotypes from the population of interest.

## Algorithm

At each step of the algorithm, a coalescence, mutation or recombination event is inferred from the existing pool of haplotypes. Each event results in a *transition* from $S_T$ to $S_{T+1}$ (note that $T$ is incremented by moving backwards in time). This iterative process continues until $S_T$ consists of only a single sequence, representing the common ancestor of all of the "contemporary" haplotypes in $S_1$. The events and resulting transitions are defined as follows:

### Coalescence

A coalescence event may be inferred when $S_T$ contains two or more sequences $C_{1:k}$ $(k \geq 2)$ such that all consist entirely of a perfectly shared tract ($[C_1, \ldots, C_k][1, m]$). All such sequences may then be "condensed" into a single parental sequence $C'$, where

$$C'[i] = \begin{cases} \bullet & \text{if all } C_{1:k}[i] = \bullet \\ \text{consensus allele at SNP } i & \text{otherwise} \end{cases}$$

The coalescence transition is then

$$S_{T+1} = S_T \setminus \{C_1, \ldots, C_k\} \cup \{C'\}$$

Intuitively, coalescence events replace "sibling" haplotypes with their "parental" haplotype, representing the process of reproduction.

### Mutation

A mutation event may be inferred when $S_T$ contains a sequence $C_M$ such that for some SNP $i$, $C_M[i] \in \{0, 1\}$ and $C_M[i] = \neg\, C_k[i]$ for all $C_k \in S_T \setminus \{C_M\}$ such that $C_k[i] \in \{0, 1\}$. The lone allele is assumed "mutant" and may be reverted back to the consensus allele for that SNP, producing a parental haplotype $C'$, where

$$C'[i] = \neg\, C_M[i]$$
$$C'[j] = C_M[j] \text{ for all } j \neq i$$

The mutation transition is then

$$S_{T+1} = S_T \setminus \{C_M\} \cup \{C'\}$$

Intuitively, mutation events resolve the "error" of a mutation which occurred at time $T$ in $C_M$'s parent.

### Recombination

A recombination event may be inferred when the criteria for coalescence and mutation events are not satisfied. Pairs of haplotypes in $S_T$ which exhibit long shared tracts are preferred candidates for recombination, since long shared tracts are assumed to have arisen from recombination relatively recently in the population's history (shared tracks generated by older recombination events are more likely to have been broken up into smaller tracts by more recent mutation or recombination events). Denoting the *recombination breakpoint* occurring between two SNPs $\alpha$ and $\beta$ as $(\alpha, \beta)$, the ideal selection of haplotypes $C_1, C_2$ is those which exhibit a relatively lengthy shared tract $[C_1, C_2][a, b]$ such that recombination produces "parents" which can then be coalesced with other haplotypes in $S_T$. The possible recombination breakpoints are $(\alpha, \beta) = (a-1, a)$ and $(\alpha, \beta) = (b, b+1)$ (note that these require $a > 1$ and $b < m$, respectively). The "recombinant" haplotype $C_R$ may be selected randomly from $\{C_1, C_2\}$. "Parental" haplotypes $C'_1, C'_2$ are then constructed, where

$$C'_1[i] = \begin{cases} C_R[i] & i \leq \alpha \\ \bullet & i > \alpha \end{cases} \qquad \text{and} \qquad C'_2[i] = \begin{cases} \bullet & i < \beta \\ C_R[i] & i \geq \beta \end{cases}$$

The recombination transition is then

$$S_{T+1} = S_T \setminus \{C_R\} \cup \{C'_1, C'_2\}$$

Intuitively, recombination events account for "crossing over" between two distinct genetic lineages over time. Note that if both $a > 1$ and $b < m$ (i.e., both recombination breakpoints are valid), $S_{T+2}$ may then be constructed similarly using the other breakpoint.

## Fine Disease Mapping

Once an ARG has been constructed, it can be used for *fine disease mapping*, or the ascription of disease causation to a select allele or multiple alleles. Putative causative alleles can be identified as those whose mutation best segregates "unaffected" and "affected" individuals; in many cases groups of "affected" individuals will cluster under a subtree of the ARG whose mutant or recombinant "parent" node can be inferred as the historical source of the trait. Ideally, select nodes of the ARG will perfectly segregate the two classes of individuals; however, due to real-world complexities of trait expression and algorithmic stochasticity, this is rarely the case. Chi-squared tests may be used to supplement ARG analysis in identifying likely causative mutations; additional heuristics may be employed to account for the variability in haplotype data and trait expression. Since there are in theory infinitely many possible ARGs for a given set of haplotypes, it can be useful to construct many plausible ARGs using the Minichiello-Durbin algorithm and average statistical results over the entire sample space in order to determine the most likely causative allele(s).

# Aside: Viral Immunology

**Peptides** are short protein subsequences, generally composed of $9-15$ amino acids.

**Proteins** only contain a subset of all possible peptides. As a result, an organism's immune system can memorize the peptide signature that distinguishes *self* peptides from *non-self* peptides.

**T cells** are immune cells with receptors that can recognize peptides. T cells can check that other cells in the body have not been infected with pathogens by examining a sample peptide from a protein degraded in that cell's endoplasmic reticulum (ER). If the T cell recognizes that the presented peptide is a non-self peptide, it can instruct the cell to kill itself, a process which helps ensure that all living cells in the body remain healthy.

**Epitopes**, or *antigens*, are short amino acid sequences that have immunogenic activity. An important problem in computational biology is predicting the epitopes present in a protein, based on its sequence and three-dimensional structure.

**Viruses** cannot survive on their own in the environment and must invade other cells to survive. Once inside a cell, viruses "hijack" the cell's own machinery to replicate themselves. The replicated viruses can then exit the cell and infect other cells; bodily functions like coughing expel the virus from the bodies and contribute to spreading the virus to uninfected individuals. The cellular machinery employed in viral replication includes *restriction enzymes*, an essential technology used in modern molecular biology.

**SARS-CoV-2** (the virus responsible for COVID-19) has a 30,000 base-pair genome which contains only a few genes. It is referred to as a *novel* coronavirus because no human has previously been exposed to it, and therefore the immune system does not yet have antibodies against it. Understanding and developing treatments for newly emerging viruses is a computational challenge that involves many algorithms discussed in CS 181 and CS 182. Important steps in the process include sequencing and assembly of the viral genome, as well as alignment of the viral genome to other known genomes. Once regions of similarity to other viruses are identified, existing drugs and vaccines can be modified to treat the new virus more effectively.

**Computational biology** plays a critical role in the development of cures for novel diseases!