
Ancestral Recombination Histories for Error Detection in Genome Sequencing



Chris Campbell, Zi Wang and Yu Qian
University of Oxford
Department of Statistics

August 13, 2010

Abstract

We first analyse and test three methods of inferring ancestral recombination histories on a simulated data set. We test the methods on a range of recombination rates and on different data set sizes. We then develop an error detection program, implementing the most successful algorithm to detect sequencing errors in genome-wide data from 19 inbred *Arabidopsis thaliana* lines.

Contents

1	Introduction	4
1.1	Motivation	4
1.2	Biological Background	5
1.2.1	Mutation	5
1.2.2	Recombination	5
1.2.3	Coalescence	5
1.3	Ancestral Recombination Graphs	5
2	Survey of Literature and Algorithms	7
2.1	Development of ARG's	7
2.1.1	Minimum Recombination Histories	7
2.1.2	Bounds on minimum recombination number	7
2.2	Newick Format	8
2.3	beagle and kwARG	8
2.3.1	Branch and Bound	9
2.3.2	beagle	9
2.3.3	kwARG	9
2.4	mARGarita	10
2.5	Blossoc	10
2.6	Metrics on the Space of Tree Topologies	11
2.6.1	Quartet Distance	11
2.6.2	Subtree-Prune-and-Regraft	11
2.6.3	Robinson-Foulds Distance	11
2.7	Results and Analysis of Testing ARG Algorithms on Simulated Data	12
2.7.1	Accuracy	12
2.7.2	Analysis	12
3	Error Detection in Arabidopsis thaliana	15
3.1	Materials and Methodology	15
3.1.1	An Initial Look at the Data	15
3.1.2	Estimation of Recombination Rate	15
3.1.3	The Ideal "Window" Size for Local Genealogy Tree Analysis	17
3.1.4	Error Detection: A Naïve Approach	18

3.1.5	Methods of Approach	19
3.2	An estimate on the number of sequencing errors	23
3.3	Extensions	24
3.4	Acknowledgements	24
A	Accuracy of Differing values of τ	27

Chapter 1

Introduction

1.1 Motivation

One of the promises genetics has made since its inception was the ability to determine genetic causes and associations with disease. For many diseases this has already come to fruition, especially those caused by single mutations at single loci, that have high penetrance (probability of presenting given a mutation). However, for a wide range of diseases, multiple mutations at several different loci may be necessary, or the mutations have low penetrance or the mutation can occur in several places but only one is necessary for a "case" phenotype, making it much more difficult and costly to determine such an association.

With the advent of cheap and efficient genomic sequencing techniques, large amounts of genome-wide data is available for discovering complex disease associated haplotypes [1]. This has led to many different association mapping techniques being developed, some involving the use of ancestral recombination graphs (ARGs hereafter, see Biological Background for definition) [4]-[8].

The Mott Group, based at The Wellcome Centre for Human Genetics, have developed 19 inbred strains of *Arabidopsis thaliana*. In order to perform Quantitative-Trait-Locus (QTL) mapping and association mapping on the descendants of said strains a high coverage sequencing was undertaken of the founder ancestries. With *de novo* sequencing at its current stage, the group require software to discover sequencing errors in the data.

Our method of approach (see 4.4) depends intrinsically on fast calculation of accurate ARG's over varying interval sizes in the data. It is therefore necessary for us to carry out an empirical test of the different algorithms available for determining ARGs on simulated data sets. This will allow us to choose the best for our purposes, and implement that algorithm in our program.

1.2 Biological Background

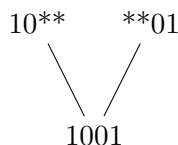
1.2.1 Mutation

Mutations are random changes between generations in the genome at a single locus. For our model we take the infinite-sites model, a well known, and often used model in bioinformatics [3]. This effectively constrains the number of point mutations to one per site maximum, and makes the simulation much easier.

1.2.2 Recombination

Recombination occurs during meiosis, for example in the formation of gamete cells in humans. Each person contains two versions of each chromosome (except sex chromosomes in males) and when a gamete cell is produced, it only contains one. This copy can be an exact copy of either of the parent chromosomes (give or take some point mutations) but can also be part of one and part of the other. This is called a crossing-over recombination and means that a sequenced haplotype may have 2 ancestors. As an example consider Figure 1.1, a simple example of one sequences recombination history where * represents an unknown historical data point.

Figure 1.1: A simple possible recombination history.



1.2.3 Coalescence

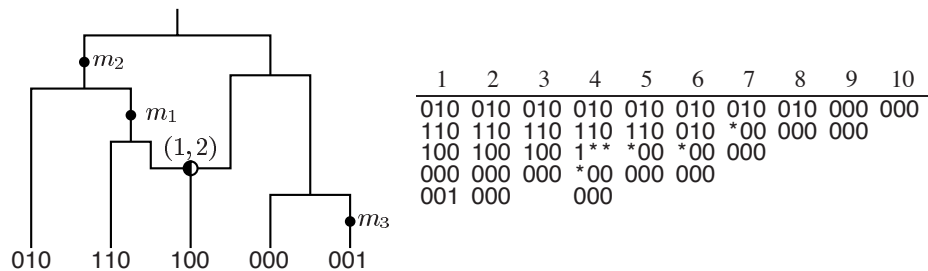
The theory of the coalescent is a mathematical model of population genetics. It is a retrospective model, that uses a sample of observed haplotypes to infer a historical genealogy or "coalescent". In our setting, a coalescent event is a possible historical event where a haplotype is split into two identical daughter copies of itself, which then go on to evolve independently. For example, allopatric speciation will occur when two copies of a haplotype are separated by geography, and when viewed retrospectively we call this a coalescent event.

1.3 Ancestral Recombination Graphs

An ARG is a graph containing all of a set of sequences' hypothetical recombinative and mutative history. Formally, an ARG is a network N that has as leave nodes as elements of a set M of observed sequence data. Say $|M| = n$ and each sequence has m positions. Then N is a directed acyclic graph containing exactly one node (the most recent common ancestor or MRCA) with no incoming edges. Every other node has precisely one or two

incoming nodes. If a node has two incoming edges it is called "recombination node" [7]. For each site from 1 to m , we can construct marginal trees, that is subgraphs of N that are trees made up of only the historical mutations and coalescences [14] of that particular site. For example, consider Figure 1.2.

Figure 1.2: A simple data set and one possible ARG (quoted from Lyngsø et al. (2005) [2]). On the left is the ARG itself, whilst on the right is a table showing the history ancestral states at each previous generation, generation 10 being the MRCA.



Of course there will be many possible ARG's for any given M . The problem lies in finding accurate or even plausible ones, and knowing how accurate or plausible your estimates are.

Chapter 2

Survey of Literature and Algorithms

We present a brief overview of the important developments in the field as it relates to our study.

2.1 Development of ARG's

Very similarly to the development of phylogenetic inference, many methods have been explored in ARG's. In the following let M be the set of observed sequences represented in binary, with all 0's being the ancestral haplotype.

2.1.1 Minimum Recombination Histories

The equivalent of maximum parsimony, minimum recombination histories are at least a plausible place to begin. We call any ARG with the minimum number of recombinations a "minARG" [9], and the minimum number itself is called $R_{min}(M)$. $R_{min}(M)$ has been shown to be NP-hard to compute [10], and given the number it has been shown to be NP-hard to reconstruct a minARG from M [11]. It therefore became important to find efficient ways of approximating this number.

2.1.2 Bounds on minimum recombination number

Several lower bounds have been discovered, with varying accuracy and efficiency to compute.

Hudson-Kaplan bound

This bound uses the infinite-sites modelling assumption from the offset. If we take any 2 site long subset of sequences from our data, we can determine if a recombination has occurred between the sites by simply noting the presence all four possible haplotypes 00, 01, 10 and 11. This implies a recombination must have occurred between these sites

as multiple mutations cannot occur at the same site [12]. This is known as the 'Four Gamete test', and is often shortened to the 'Three Gamete test' if the ancestral genome is known.

Haplotype bound

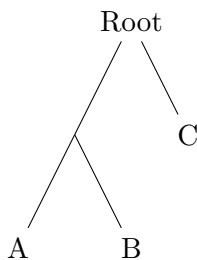
A simpler to compute, but weaker bound can be calculated simply by making a few simple observations. The historical events that are allowable in the model are mutations, coalescences and recombinations. Clearly coalescent events do not change the number of distinct haplotypes and so if we have H distinct haplotypes, and $E = R+S$ is the number of recombination events plus the number of mutation events, then we have $E \geq H - 1$. This gives a bound of

$$R_{min}(M) \geq H - S - 1$$

Although this bound can be negative, in many situations it is a perfectly useful starting point.

2.2 Newick Format

We use a standard format for computing with trees called Newick format. This is based around a shared characteristic of trees with nested sequences of bracketed pairs. For example, take this rooted tree.



In Newick format this would be represented as $((A,B),C)$. Each bracketed pair represents a parent node with the two children nodes being named in the brackets. This is a useful and efficient method of storing and using tree data, and has been adopted widely.

2.3 beagle and kwARG

Developed by Lyngsø et al. [2], beagle is a branch and bound algorithm, proven to construct a minARG given sufficient time. However, on the data sets we used, beagle would take many months of computer hours to calculate the ARG's so we use beagle's heuristically improved younger brother, kwARG, also developed by Lyngsø.

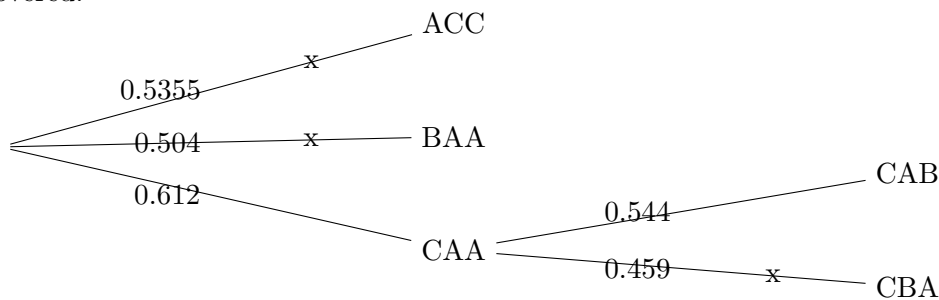
2.3.1 Branch and Bound

Branch-and-Bound is a general algorithm used in many optimisation problems. As an example we take a maximisation problem. Maximise the probability of winning a game where you must choose an order for three players to attempt certain tasks where the probabilities of each player succeeding at each task are:

Table 2.1: Probabilities of Success

Task	A	B	C
1	0.9	0.7	0.85
2	0.8	0.6	0.7
3	0.9	0.8	0.85

A simple branch and bound algorithm would "branch" on the first task, and then "bound" by assuming we could then just use the best player for the second two games. By choosing the branch with the highest bound we would discard many possible solutions with a small number of computations, and this can be recursively run until a solution is discovered.



And so in this case we can see the bounding argument finds a solution of CAB and this solution is certainly the global one as it is bigger than the bounding probabilities on the other nodes.

2.3.2 beagle

The aim in beagle is to minimise the number of recombinations necessary to reach the coalescent. It 'branches' on the possible first step in the history, and for each possibility calculates the haplotype bound for $R_{min}(M)$ on the new data set. Although this process is guaranteed to terminate with a minARG, it intuitively does so slowly as before it can show an ARG exists for $R_{min}(M)$ recombinations, it must first show no such ARG exists for $R_{min}(M) - 1$.

2.3.3 kwARG

kwARG implements all of the refinements suggested in [2]. These are mainly focused around cutting down the data set and making mutation and coalescent events more

easily than recombination. kwARG allows for any initial lower-bound scoring scheme, and we use the default (haplotype bound) so as to test it under likely laboratory use conditions.

2.4 mARGarita

mARGarita essentially has the same structure as kwARG but differs on a number of key points.

1. mARGarita emphasizes on inference of plausible ARGs rather than restricting the result on minimum ones.

2. The introduction of the concept "shared tract" between two sequences. This requires:

- (1) The two sequences have the same allelic state over the shared tract. (e.g. 100 & 1*0)

- (2) At least one allelic state over the shared tract is (explicitly) defined. (e.g. ***0*1*)

- (3) The shared tract is maximal, so that inclusion of an extra allelic state at either ends will violate (1)

Note that at any stage, it is possible that more than one options from coalescence, mutation and recombination are available. We choose between these events using the following heuristics:

- (1) Mutations and coalescences always come before recombinations wherever possible.

- (2) If multiple mutations and/or multiple coalescences are both possible at the same stage, the order is determined randomly.

- (3) Coalesce sequences only if they have an overlapping region of defined material. i.e. at least one position where the 2 sequences match is not a "*".

- (4) Recombinations are added at the ends of longest shared tract with default probability 0.9, in the light of that longer shared tracts tend to arise from more-recent recombination events.

- (5) The first coalescence after a recombination is based on the shared segment used to decide the location of that recombination.

2.5 Blossoc

Blossoc (BLOck aSSOCIation) is an algorithm presented in [5] that builds local perfect phylogenies and then tests these for accuracy as decision trees for case-control status. We can use these as the marginal trees for the test. To build the perfect phylogenies a number of heuristics are employed and this results in very fast computation time.

2.6 Metrics on the Space of Tree Topologies

Many metrics have been developed for measuring the difference between tree topologies over a given data set. We present three of these and discuss which will be most useful for our requirements.

2.6.1 Quartet Distance

Quartet distance is found by comparing the topology of subtrees, each showing the relationships between every subset of the data of size 4. The distance itself is simply the number of quartets that have differing subtree topologies in each tree. Although this algorithm can be calculated efficiently, it does not take into account how distantly related pairs of haplotypes are and so throws away a lot of the important information for our task.

2.6.2 Subtree-Prune-and-Regraft

If we define an SPR move to be any cut made along an edge of a tree and then a re-grafting onto another edge. For example consider Figure 2.1, where a cut is made disconnecting the subtree with 1 and 2 and this subtree is re-grafted onto the edge ending with 6. The SPR distance is defined as the minimum number of SPR moves it takes to convert one tree to another.

Unfortunately, no computationally efficient method exists for calculating this distance and so we cannot use it in our experiment.



Figure 2.1:

2.6.3 Robinson-Foulds Distance

Robinson-Foulds distance (RF hereafter) is a metric on the space of trees that uses partitions between data points in the set to measure differences in trees. Say we have two trees (N_1, e_1, V_1) and (N_2, e_2, V_2) , for each vertex $v \in V_i$ define the cluster of v as the set of leaves descended from v . Now define

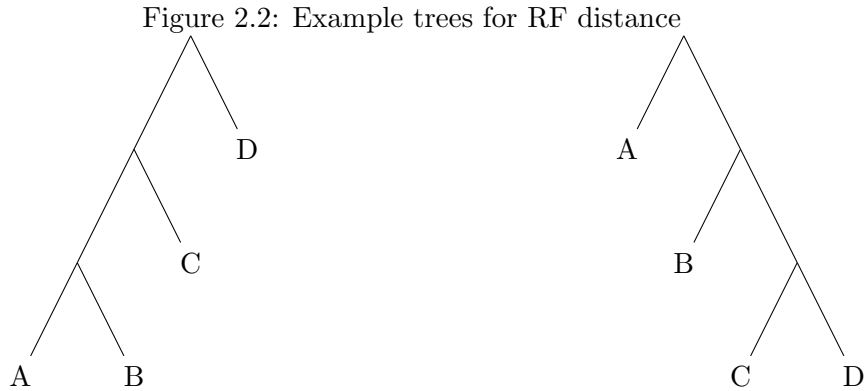
$$C(N_i) = \{C(v) : v \in V_i\}$$

Then the RF distance between the trees is

$$d_{RF}(N_1, N_2) = \frac{|C(N_1)C(N_2)| + |C(N_2)C(N_1)|}{2}$$

For example, if we have two trees as in Figure 2.2, we can calculate their RF distance by adding these numbers. By inspection, we can see that the RF distance in this case is 2, as a partition between A and B is swapped to a partition between C and D.

RF distance is the most suited to our requirements as it is a measure of the differing distance of relationship between pairs of haplotypes and has also been implemented efficiently in HashRF [13] which runs in linear time in the number of leaves.



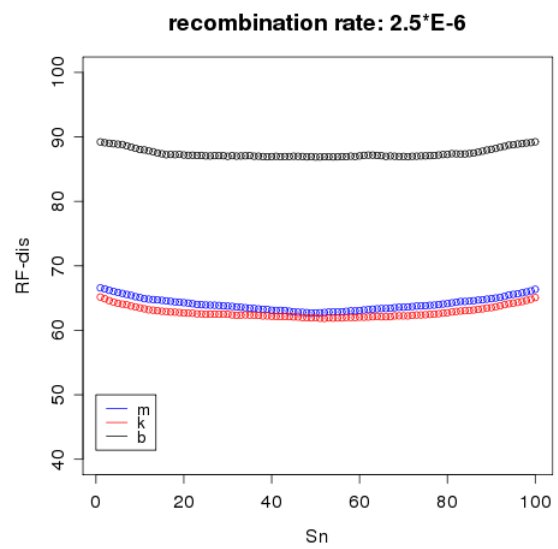
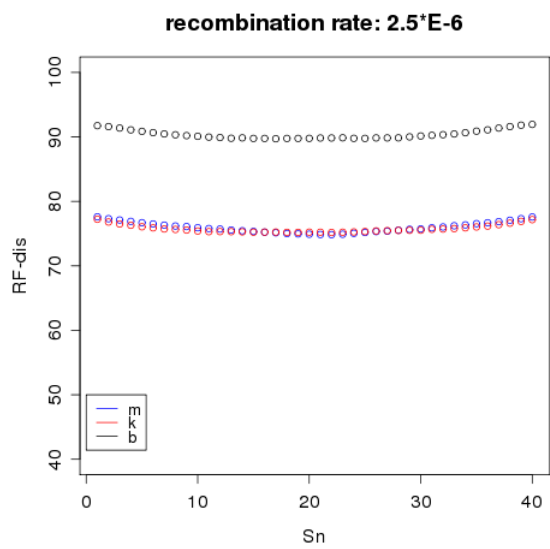
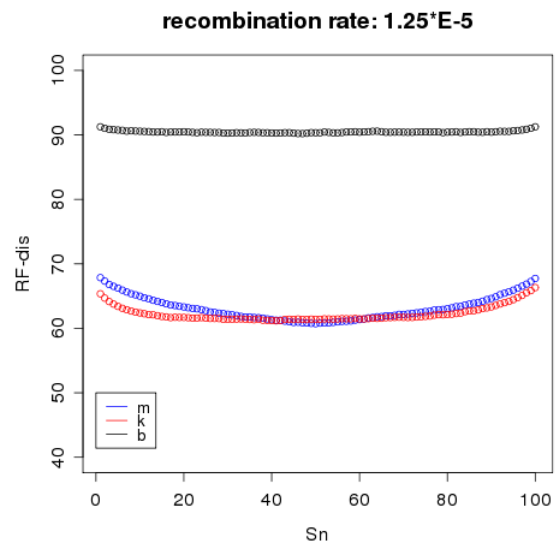
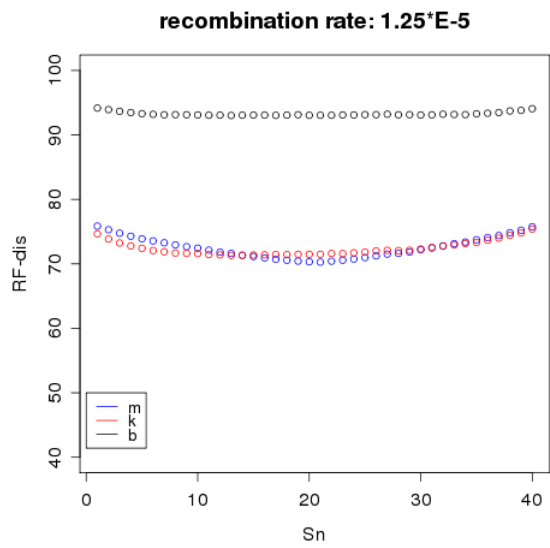
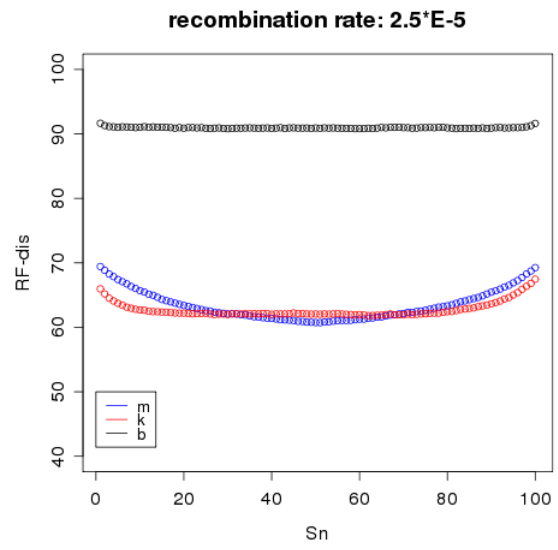
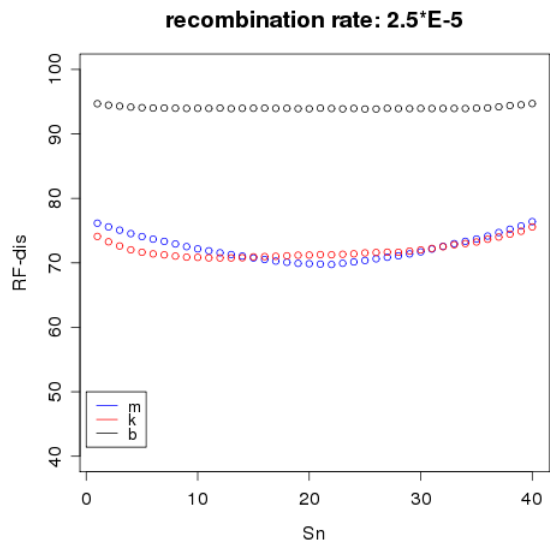
2.7 Results and Analysis of Testing ARG Algorithms on Simulated Data

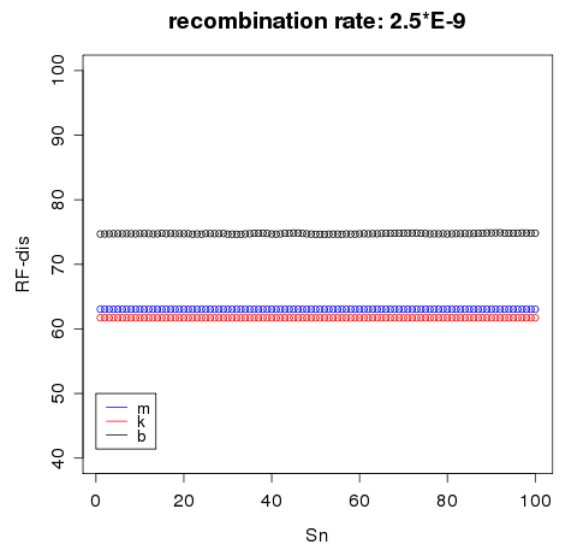
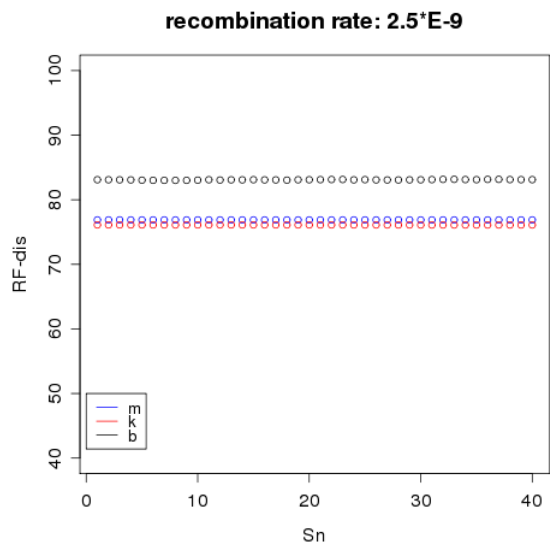
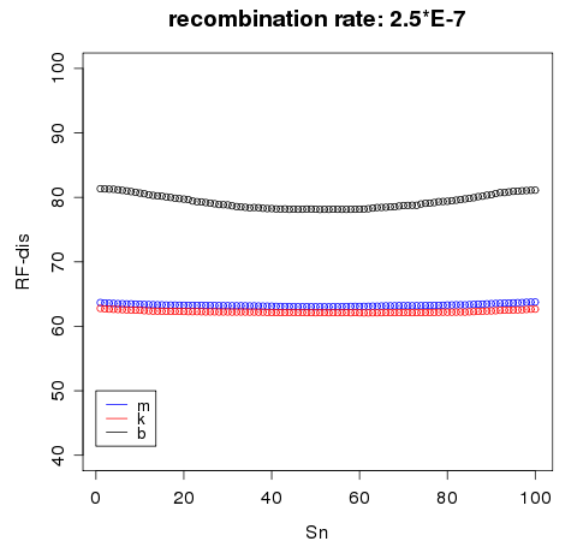
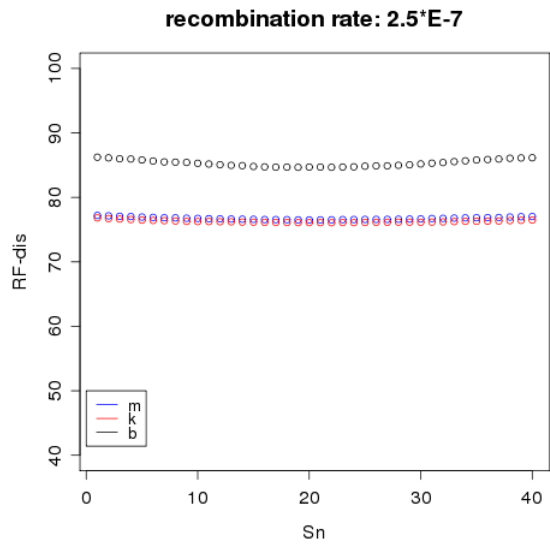
2.7.1 Accuracy

Below we present the results of running each algorithm on 10 simulated data sets, each having either 40 or 100 segregating sites and varying recombination rates. From the definition of RF-distance, a higher distance is worse so lower marks are given for better algorithms.

2.7.2 Analysis

Our requirement for the next section of is fast computation with accuracy. Although kwARG consistently outdoes mARGarita in accuracy, a first look estimate of running time for kwARG on the Arabidopsis genomic data is 263 days, whilst mARGarita would only last 2.3 days. So in moving forward we decided to use mARGarita in the next stage of the project.





Chapter 3

Error Detection in *Arabidopsis thaliana*

3.1 Materials and Methodology

3.1.1 An Initial Look at the Data

The *Arabidopsis thaliana* data consist of 5 chromosomes, each with 19 genomes, one of which being the reference. We firstly transform the data into binary form by alignment with the reference genome: 0 representing ancestral state, and denote by 1 otherwise.

As all the methods involved in the following study are primarily based on observations of recombination histories, we can speed up the procedure by removing the uninformative sites (sites where at most one genome carries a mutant type [2]) without affecting the main results derived using our method. 3.1.1 gives a summary of total number of segregating sites and informative sites in each of the 5 chromosomes.

Chromosome	Segregating Sites	Informative Sites	% Informative
1	768885	419587	55.4
2	569132	300582	53.4
3	672192	355676	53.6
4	557951	297797	54.1
5	696606	392667	56.4

Figure 3.1: Table showing informative sites

From now on, the term AT-data will refer to this data set consisting of informative sites only.

3.1.2 Estimation of Recombination Rate

To best reflect the characteristics of the *Arabidopsis thaliana* data in the use of simulation, we start by inferring the recombination rate of the data, assuming the rate is

constant across the whole region. It follows from coalescent theory that population samples contain information on the value of the product of the recombination rate c and the effective (diploid) population size N , but not on c and N separately. It has therefore become standard to attempt to estimate the compound parameter $\rho = 4Nc$ [15].

For a fixed value of ρ , we use `mksample` [16] with this parameter to generate 18 random sequences and add a reference sequence (only consists of a string of zeros), all of which have length 200. We then remove the uninformative sites so that the sequence lengths are reduced to a range around 100 (more than 50% of sites are discarded). Assuming a linear relationship between HK-bound and the number of informative sites, we rescale the simulation results of HK-bound as for sequences with length 100. There are three easy ways to evaluate recombination rate haplotype bound, HK-bound and number of recombinations predicted by `kwarg`. To save computation time, we use the HK-bound as rough evaluation of our simulations.

In order to compare with real data, we randomly draw 896 samples (consists of 19 sequences and 100 columns) with length 100 from AT-data, and plot the distribution of both HK-bound and number of recombinations predicted by `kwarg`, shown as Figures 4.6-7.

Results

Figures 4.2-5 show the distribution of the HK bound across a simulated data set.

Comparing with the distribution of HK-bound from real data (Figure 4.6), we identify $\rho = 200$ to be the best estimate. We exclude the plots for bigger values of ρ as we observe that the HK-bound associated with peak frequency increases as we increase ρ .

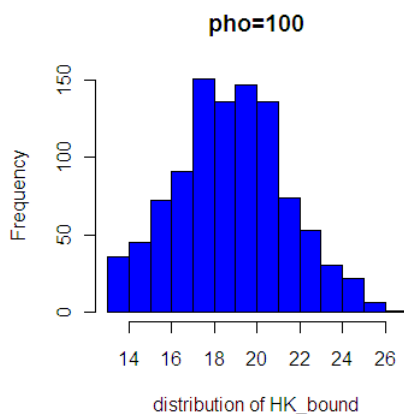


Figure 3.2:

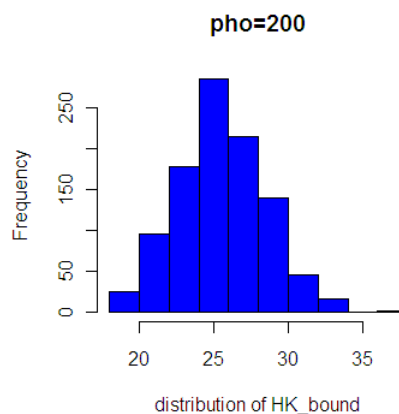


Figure 3.3:

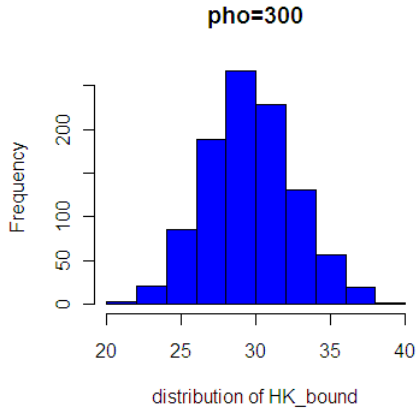


Figure 3.4:

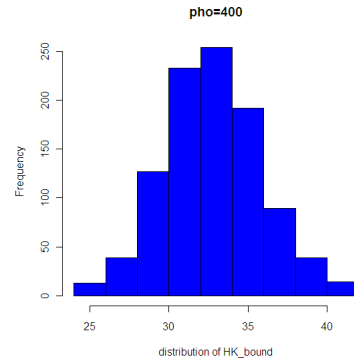


Figure 3.5:

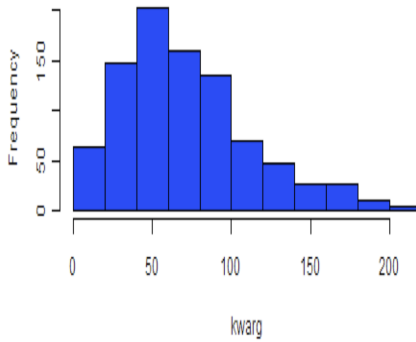


Figure 3.6: Distribution of kwARG inferred recombination numbettr in AT data

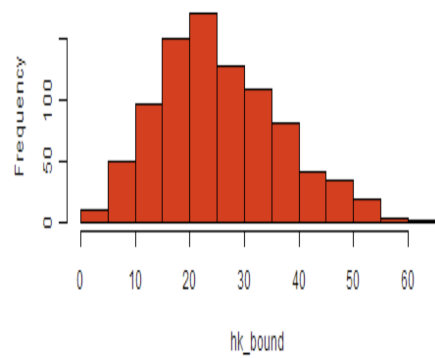


Figure 3.7: Distribution of HK bound in AT data

3.1.3 The Ideal "Window" Size for Local Genealogy Tree Analysis

Inferring sequencing error based on local genealogy trees requires a sliding window along the genome, in which we can compute the local genealogy history within reasonable computation time and with relatively good results. We use `mksample` to generate 19 sequences of length n , with recombination rate $\rho = 200$. By doing this, we have a record of the true genealogy of the entire window. τ is chosen from 50 to 800 with an interval of 50. For each candidate τ , 100 iterations are made by using `mARGarita` to construct marginal trees for each of the τ sites, RF-distance with respect to the true trees and running time are recorded. (Table 3.1)

Results

For plots of mean RF for each site on various fixed values of τ , see appendix. (16 plots)
We observe that:

- (1) As τ increases, mean RF drops from around 14 to just below 10.
- (2) There isnt much difference between the plots for $\rho = 500$ and $\rho = 800$.
- (3) All the plots have a U shaped pattern, due to more information gathered for the sites in the middle. Yet the pattern is not striking in any particular plot.

τ	Total time(second)	Time per site
50	22	0.44
100	27	0.27
150	33	0.22
200	40	0.2
250	46	0.18
300	54	0.18
350	61	0.17
400	69	0.17
450	72	0.16
500	79	0.16
550	89	0.16
600	97	0.16
650	105	0.16
700	115	0.16
750	125	0.17
800	135	0.17

Table 3.1: Table Showing Running Time Against τ , the Window Size

We discover that as τ increases, the variable time per site decreases and seems to converge to 0.17, hence we may assume a near-perfect linear relationship between τ and total time.

Combining the results we get from the plots and the table, we conclude $\tau = 50$ would be a satisfying window size, although for the more concerned, $\tau = 200$ or larger might be more preferable. Yet theres no point in going beyond $\tau = 500$ which will add in little influential information but increase the total running time. We also recommend using marginal trees constructed at a portion (say 1/3, from a conservative view) of the central sites in each window as reliable inference due to the variability of RF displayed.

3.1.4 Error Detection: A Naïve Approach

Based on the fact that recombinations are flanked by two incompatible regions of the genome whereas errors are unlikely, we propose the following methods as a start point for error detection.

3.1.5 Methods of Approach

We found several methods for error detection that all follow the same basic pattern. A window moves along the data, with one column at its centre. Calculations (see below) are made based upon the interval with and without this central column included (see figure below). By comparing these calculations, we should be able to accurately determine whether or not an error has occurred.

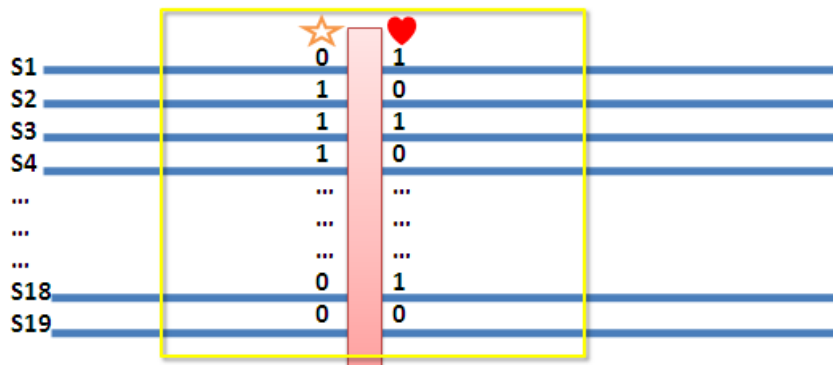


Figure 3.8: Window moving along the data. The heart and star represent different marginal tree topologies at those sites. This would imply an error is less likely than a recombination

Strain Distribution Patterns

In [17], use was made of the structure of the strain distribution patterns (SDPs) to determine sequencing errors. In this paper, the data had 13 prominent SDPs, making up 98% of the data. A brief survey of our data found 500 sequences making up only 40% of the data. This means that any test based upon this method would be computationally unviable and so we must look elsewhere for ideas.

HK Bound

By calculating the HK bound on the interval with and without the central column you might expect a significant change in this bound. We ran an implementation of this algorithm over a simulated data set and found very little sensitivity to errors and so had to abandon this method.

Haplotype Bound

By the same reasoning as above, we hoped to discover errors using the haplotype bound. Unfortunately, this led to a very computationally inefficient algorithm, requiring many weeks of computation for a data set of the size we have, and so this method too failed to be useful for our requirements.

Tree Topology

By calculating the marginal trees of sites within the window we expected to see drastic changes in tree topology. This method was also insensitive to errors and so had to be abandoned.

Worked Method 1

Firstly, we fix the size of a detecting window, denoted by τ . At the x -th column, we explore the $2 * \tau$ columns numbered $x - \tau, x - (\tau + 1), \dots, x - 1, x + 1, \dots, x + (\tau - 1), x + \tau$. (We denote this set by $\gamma(x)$). If every pair of columns from $\gamma(x)$ are compatible (i.e. where we observe the presence of at most 3 haplotype types from 00, 01, 10, and 11), yet the $(x-1)$ -th and $(x+1)$ -th columns are incompatible with the x -th column, we conclude the x -th site contains at least one sequencing error.

Worked Method 2

We loosen the condition on compatibility in $\gamma(x)$ in method 1 as following: instead of requiring the neighbouring sites of the x -th column to be incompatible with this column, we regard a sequencing error sitting in column x if there exists one site in each of its left and right τ neighbouring sites that is incompatible with the x -th column. This addition does not violate with our guidance principle at the beginning of this sub-section, while it has an advantage over method 1 in detecting sequencing errors that are contained in neighbouring columns.

Results on Simulated Data

To evaluate the performance of our algorithms in 3.1.5, we firstly display the results from simulation. The simulated data consist of 19 genomes, one of them being the ancestor with 0s throughout, each of length 1000. They are generated using `mksample` with recombination parameter $\rho = 100, 200, 300, 400$ respectively, which we shall deal with in separate groups. 15 sequencing errors are added at random (distinct) positions by flipping a 1 into 0, or vice versa, and for each value of ρ , we iterate the process for 1000 times.

In our simulation, we record the total number of calls (TC, which is the total number of inferred sequencing errors), the number of calls that are actually not sequencing errors as we set at the beginning (false positive = FP) which we didnt display in the tables. And we derive the confidence level (C.L.) of our calls by defining: $C.L. = \frac{TC - FP}{TC}$

A larger value of this means we have better confidence that the call made is in fact a sequencing error.

We define the Power of an algorithm to be the probability an error is called given it is there. $Power = \frac{TC - FP}{15000}$, As there are $15 * 1000 = 15000$ sequencing errors in each test consisting of 1000 iterations. A larger value of this suggests that an algorithm has better capability of inferring sequencing errors.

Method	ρ		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 10$
Method1	100	Total Calls	1596	1364	1141	903	700	175
		C.L.	0.92	0.95	0.97	0.98	0.98	0.98
		Power (%)	9.82	8.65	7.38	5.87	4.59	1.15
	200	Total Calls	1864	1376	978	689	477	53
		C.L.	0.82	0.88	0.92	0.93	0.94	1.00
		Power (%)	10.19	8.11	5.98	4.27	2.98	0.35
	300	Total Calls	1961	1243	752	457	280	14
		C.L.	0.69	0.78	0.83	0.86	0.88	0.86
		Power (%)	9.05	6.47	4.18	2.61	1.64	0.08
	400	Total Calls	2260	1299	715	381	190	3
		C.L.	0.60	0.71	0.77	0.80	0.87	1.00
		Power (%)	9.00	6.14	3.68	2.04	1.11	0.02
Method 2	100	Total Calls	1596	3200	3922	3925	3563	1179
		C.L.	0.92	0.92	0.90	0.88	0.87	0.85
		Power (%)	9.82	19.57	23.55	23.15	20.59	6.65
	200	Total Calls	1864	3463	3760	3366	2615	359
		C.L.	0.82	0.77	0.74	0.72	0.71	0.66
		Power (%)	10.19	17.72	18.47	16.18	12.41	1.58
	300	Total Calls	1961	3572	3563	2813	1951	140
		C.L.	0.69	0.64	0.60	0.57	0.56	0.56
		Power (%)	9.05	15.15	14.20	10.62	7.23	0.53
	400	Total Calls	2260	4034	3649	2586	1618	59
		C.L.	0.60	0.53	0.50	0.48	0.47	0.41
		Power (%)	9.00	14.36	12.21	8.28	5.09	0.16

Table 3.2: Number of errors inferred in simulated data under the two methods. Total calls is cumulative number of errors detected, C.L. is confidence level, and power is the percentage of errors detected.

Conclusions

Within parameters range as included in simulation:

1) Method 1 always gives more inferred errors than Method 2, unless we restrict the size of detecting region (τ) to one where methods 1 and 2 are the same- this is to be expected, as we loosened the condition of inferring an error in Method 2.

2) The confidence level of our inference is much higher using Method 1 under same parameters. In particular, for the most realist parameter $\rho = 200$, $\tau = 3$ will give confidence level of 0.92, a very desirable result. Whereas Method 2 gives confidence level of 0.74.

3) The power of both algorithms is extremely low, in particular Method 1. This is because sequencing errors do not necessarily conflict with immediate neighbours. Moreover in regions with high recombination rate, our method has significantly lower power as recombinations add many incompatibilities. In the light of this observation, we compute

another simulation with recombination parameter $\rho = 0$, i.e. no recombination events. The results are displayed in the following table:

Method	ρ		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 10$
Method 1	0	Total Calls	1542	1495	1449	1383	1339	1119
		C.L.	0.99	0.99	0.99	0.99	1.00	1.00
		Power (%)	10.21	9.94	9.65	9.21	8.93	7.46
Method 2	0	Total Calls	1542	3340	4640	5429	6054	6605
		C.L.	0.99	0.99	0.99	0.99	0.99	0.99
		Power (%)	10.21	22.22	30.87	36.13	40.30	44.03

We read from the table and infer that the power of the algorithm can be at most around 0.10 using method 1.

As a final remark, we realize that dealing with the issue of recombination hotspot may be one of the desirable future directions. This will be discussed in 3.3, and [18] has more on this topic.

Results on AT Data

Then we applied the algorithms to the real data. The following table shows the number of columns with potential sequencing error for each chromosome in the AT-data based on our method 1.

Chromosome \ τ	2	3	4	5	10
1	3518	1732	1004	617	128
2	2728	1236	670	395	70
3	3007	1374	758	480	86
4	2353	1085	553	340	53
5	2924	1391	799	506	101
Total	14530	6818	3784	2338	438

A more interesting question arising from this result would be: how many of these errors (in percent) can be explained by correcting only one figure in each column? We compute the answer by flipping each number in turn in each column where we believe there are sequencing errors, and we obtain the following results using method 1:

We see about 70(%) of the inferred error containing sites can be explained by changing just one string at that site(i.e. by changing one character in that column) (See Table 3.3).. It is important to note that although only one change is needed to remove the incompatibility, it may be that many strings have this property at each site.

Chromosome \ τ	3	4	5	10
1	67.4	70.2	73.6	80.5
2	65.4	66.7	70.6	71.4
3	66.2	69.4	71.7	76.7
4	64.3	66.0	68.5	67.9
5	63.6	66.5	68.8	74.3

Table 3.3: Table showing percentage of errors that are removed by only one character change in the column

3.2 An estimate on the number of sequencing errors

In the above analysis, we assume from past experience that the sequencing error rate is about 0.0010-0.0015. And set the number of sequencing errors or other related parameters in the relevant study (for example, in simulation and algorithm tests) accordingly. This illustrates the demand for an easy-to-compute and yet precise way of estimating the number of sequencing errors (hence its rate) in a given set of data. We turn to a statistical approach to this problem.

Motivated by the facts:

1. # (natural) mutations $\propto \log(\# \text{ sequences in the set})$
2. # sequencing errors $\propto \# \text{ sequences in the set}$

We construct a linear regression model to find the estimate. Let the index $[i]$ denote the label of a set of sequences, so i takes values 1, 2, 3, ... Let $S[i] :=$ number of sequences for the set labelled i , and always assume $S[i]$ bigger than one. Let $n[i] :=$ number of segregating sites in the set labelled i . Using R, we find the coefficients a , b , and c in the linear model:

$$n = a + b \log(S) + cS$$

with each i corresponding to an observation: $n[i]$, $S[i]$, and the derived $\log(S[i])$.

Taking our AT-data as an example, $S[i]$ will be a number in $\{2,3,4,\dots,19\}$, and i will be an integer between 1 and $(2^{19}) - 19 - 1$.

Once we find the coefficients, in particular, b . We can consequently make the estimation of the number of sequencing errors by $b \log(\text{number of sequences})$.

The tables below show the minimum, maximum and mean number of segregating sites observed in different sizes of subsets of the AT data. [h!]

Using mean values only, computing with R, we obtain $b = 838746$ in which case, for 19 strains, we expect $37364 * 19 = 709916$ errors. Our program detected 14530 errors,

Statistic \ Number of Strings	2	3	4	5	6	7	8	9	10
Minimum no. of Seg. Sites	428511	877354	1135429	1343925	1531289	1699111	1859551	2010079	2148350
Mean no. of Seg. Sites	752217	1128325	1401865	1624120	1814389	1982321	2133545	2271676	2399202
Maximum no. of Seg. Sites	921234	1339544	1625554	1848617	2039802	2200712	2346379	2475853	2595769

Statistic \ Number of Strings	11	12	13	14	15	16	17	18	19
Minimum no. of Seg. Sites	2273978	2394583	2506046	2616787	2725419	2828926	2936884	3060624	3264761
Mean no. of Seg. Sites	2517915	2629162	2733981	2833195	2927466	3017339	3103267	3185632	3264761
Maximum no. of Seg. Sites	2706864	2809885	2900693	2986059	3065326	3137374	3199239	3241363	3264761

reflecting the low power of the current stage of this test.

3.3 Extensions

This project has made a fairly simplistic attack on the problem of error detection, and much more powerful approaches could be developed without too much extra work. One example is the introduction of a dynamic window size. If the minor allele frequency (MAF) of contiguous sites about a site is low, then the chances of the interval failing the four-gamete test are lowered, regardless of the truth of the read. In such circumstances a larger window size would be suggested, whilst if the MAF is high, then the computation over a large interval becomes wasteful and so slows the program down unnecessarily.

Therefore it would be sensible to introduce a quick MAF count around a given site before any bounds or ARGs are calculated, and then this number could be used to determine the size of the window that will give an accurate test for an error in that site. This would both increase the accuracy and decrease the running time of the program.

Another important improvement would be a testing criteria for recombination hotspots. Our method will struggle to correctly identify errors in the presence of a high recombination rate, and so recombination hotspots will cause intervals of low confidence in the test. If the algorithm could first identify hotspots and remove them from the test, and then use a different error detection method on those areas then it would be significantly more powerful.

3.4 Acknowledgements

We would like to thank Rune Lyngsø, Jotun Hein, Adam Novak and Richard Mott for all of their help, patience and ideas!

Bibliography

- [1] International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7603):1299-1320, 2005
- [2] Rune B. Lyngsø, Yun S. Song and Jotun J. Hein. Minimum recombination histories by branch and bound. In Rita Casadio and Gene Myers, editors, *Proceedings of the 5th Workshop on Algorithms in Bioinformatics (Proc. WABI), number 3692 in Lecture Notes in Bioinformatics*, pages 239-250. Springer, 2005.
- [3] Kimura, M., 1969 The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics* 61:893-903
- [4] Zhihong Ding, Thomas Mailund and Yun S. Song. Efficient whole-genome association mapping using local phylogenies for unphased genotype data. *Bioinformatics*, 24(19):2215-2221, 2008.
- [5] Thomas Mailund, Søren Besenbacher and Mikkel H. Schierup. Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, 7:454, 2006
- [6] Mark J. Minichiello and Richard Durbin. Mapping trait loci by use of inferred ancestral recombination graphs. *American Journal of Human Genetics*, 79(5) 910-922, 2006.
- [7] Yufeng Wu. Association mapping of complex diseases with ancestral recombination graphs: Models and efficient algorithms. *Journal of Computational Biology*, 15(7):667-684, 2008
- [8] Sebastian Zöllner and Jonathat K. Pritchard. Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics*, 169:1071-1092, 2005.
- [9] Song, Y. and Hein, J. 2003. Parsimonious reconstruction of sequence evolution and haplotype blocks: finding the minimum number of recombination events. *Proc. WABI*. 287-302
- [10] Wang, L., Zhang, K., and Zhang, L. 2001. Perfect phylogenetic networks with recombination. *J. Comput. Biol.* 8, 69-78

- [11] Bordeqich, M. and Semple, C. 2004. On the computational complexity of the rooted subtree prune and regraft distance. *Ann. Combin.* 8, 409-423.
- [12] Hudson, R., and Kaplan, N. 1985. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics* 111: 147-164.
- [13] Sul, S. J., and Williams, T. J. 2007. A randomized algorithm for comparing sets of phylogenetic trees. *Proc. Fifth Asia Pacific Bioinformatics Conference (APBC'07)*, 121-130.
- [14] Kingman, J.F.C. 1982 On the Genealogy of Large Populations. *Journal of Applied Probability* 19A:2743
- [15] Li , N., and Stephens, M. 2003. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics* 165:2213-2233
- [16] Hudson, R. R. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18(2):337-8
- [17] Yalcin B., et al. 2004. Unexpected complexity in the haplotypes of commonly used inbred strains of laboratory mice. *PNAS* vol. 101 no. 26:9734-9739
- [18] Fearnhead, P. and Smith,. N. G. C. 2005. A Novel Method with Improved Power To Detect Recombination Hotspots from Polymorphism Data Reveals Multiple Hotspots in Human Genes. *American Journal of Human Genetics* 77(5):781-794

Appendix A

Accuracy of Differing values of τ

The following are graphs showing the effect of changing the value of τ on accuracy of ARG inferral.

