An Accurate Genetic Clock, *GENETICS*

David H. Hamilton*,¹

*Department of Mathematics, University of Maryland, College Park

ABSTRACT Our method for “Time to most recent common ancestor” TMRCA of genetic trees for the first time deals with natural selection by apriori mathematics and not as a random factor. Bioprocesses such as “kin selection” generate a few overrepresented “singular lineages” while almost all other lineages terminate. This non-uniform branching gives greatly exaggerated TMRCA with current methods. Thus we introduce an inhomogenous stochastic process which will detect singular lineages by asymmetries, whose “reduction” then gives true TMRCA. This gives a new phylogenetic method for computing mutation rates, with results similar to “pedigree” (meiosis) data. Despite these low rates, reduction implies younger TMRCA, with smaller errors. We establish accuracy by a comparison across a wide range of time, indeed this is only y-clock giving consistent results for 500-15,000 ybp. In particular we show that the dominant European Y-haplotypes R1a1a & R1b1a2, expand from c4000BC, not reaching Anatolia before c3800BC. This contradicts previous clocks dating R1b1a2 to either the Neolithic Near East or Paleo-Europe. However our dates match R1a1a & R1b1a2 found in Yamnaya cemeteries of c3300BC by Nielsen et al (2015), Pääbo et al(2015), together proving R1a1a & R1b1a2 originates in the Russian Steppes.

KEYWORDS Molecular clock ; TMRCA ; SNP ; STR ;

Introduction

Zuckerkandl & Pauling (1962) made the empirical observation that “time to most recent common ancestor” (TMRCA) of a genetic tree seemed proportional to genetic variance. Thus began the molecular clock, a recent account can be found in Wakeley(2013): a very short list of applications might be the divergence of the eukaryotes (Kimura and Ohta, 1973)... the origins of human immunodeficiency virus (Korber et al., 2000). The Y(chromosome) clock also gives an important tool of archeology since pioneering work by L.L. Cavalli-Sforza et al(1978) correlated genetic variances to age of human lineages. Although in broad agreement with the fossil record or ¹⁴C dating, very large confidence intervals meant that questions such as “peopling the Americas” or the origin of the dominant European Y-haplotype R1b1a2 could not be accurately resolved.

Moto Kimura’s theory of neutral mutations(1968) predicted a molecular clock with constant rate. This has been applied in many versions, we call KAPZ, see Nielsen(2005). Actual timing

requires calibration of mutation rates, either by counting genetic differences over a lineage, i.e the phylogenetic method, or by direct observation of parent-offspring pairs, i.e. meiosis. For the human Y-chromosome Burgella and Navascue(2011) combined 80 meiosis studies, which for our 33 markers averages at about .0043 mutations per generation, while their alternative regression method averages at .0026. Chandler(2006) combines 20 meiosis studies to get an average of .0032. Unfortunately these rates when used in KAPZ do not correlate very well with known genealogy. So FTDNA used their huge data base to give an average of .00635. Indeed for the Y-clock in general “phylogenetic” rates are about 2 times larger than those from meiosis. While this gives plausible results for genealogy it makes ancient lineages seem far too young. Zhivotovsky (2000) conceived of a much smaller “long term evolutionary rate”, computing on average .0007 from genetic lineages of remote Polynesian populations. Myres et al (2011) used this in a KAPZ giving 9000BC for the origin of R1b1a2, i.e. right after the Ice Age.

However a long term constant rate has proved untenable as estimating rates from different genetic lineages of known ages gave rates with significant discrepancies, see Nie and Kumar(2000). Its variability was attributed to changing generation times, metabolism etc. and of course natural selection, see Ohta(2002), Lynch(2010). Some sophisticated empirical

Copyright © 2015 by the Genetics Society of America

doi: 10.1534/genetics.XXX.XXXXXX

Manuscript compiled: Friday 4th September, 2015%

¹Department of Mathematics, University of Maryland, College Park, md 20742, dhh@math.umd.edu

approaches allowing variable rates have been developed, see Nielsen(2005). However confounding examples convinced Ayala(1997) to challenge the very idea of molecular clock hypothesis, reasoning that variable rates are the result of the unpredictable events of natural selection. Never the less, while future natural selection cannot be predicted, we show how singular events of the past can be detected.

Human populations have been classified by Y(chromosome) haplotypes defined by a SNP (single nucleotide polymorphism) mutation, a famous example being R1b1a2. Now one is looking at mutations of “short tandem repeat” (STR) at different DNA Y-chromosome Segments (DYS). In applications of the Y-clock researchers have stuck with constant mutation rates. However, comparing rates at the same DYS markers we see an average 50% variation over eight major Y-haplotypes, i.e the same phenomena seen for the very long term allele clock. Now as a huge variation in human generation times etc over recent time does not seem plausible researchers have fallen back on the stochastic process itself to generate unlikely distributions. These Bayesian methods use Monte-Carlo simulations of all possible genealogical trees giving the present sample data, then find TMRCA by a maximum likelihood estimate. An example of this for the Y-clock is the algorithm BATWING (Wilson et al, 2003). Now for BATWING the TMRCA is often greater than the KAPZ, e.g. for the Cinnioglu et al(2004) study of Anatolian R1b1a2 both methods were applied to the same data and mutation rates. The KAPZ gives TMRCA 9800BC compared with 18,000BC for BATWING. Balaesque et al (2010) used BATWING to give an origin for R1b1a2 in Neolithic Anatolia c.6000BC, however disputed by Busby (2012). Another Bayesian algorithm is BEAST, see Drummond and Rambaut(2007), which has been used for mitochondrial clock where finding the phylogenetic tree is easier. Unequal fertility can be represented by non-uniform branching but in Bayesian Models being less probable increases TMRCA.

Instead, we deal with natural selection by apriori mathematics and not as a random factor. Bioprocesses such as “kin selection” generate a few overrepresented “singular lineages” while almost all other lineages terminate. These singular lineages are extremely unlikely so cannot be predicted. Instead we introduce an inhomogenous stochastic process where we have constant mutation rates but variable sources representing nonuniform fertility, i.e. many virtual patriarchs giving a quasi-mixed population. To find these sources is an inverse problem, however inversion is not stable or even unique: the population could have been created yesterday! However we are saved by finding the present population is dominated by a relatively few “singular lineages”. Our model has many extinct twigs with a few successful branches. In particular about 50% of the DYS markers show a singular side branch. This requires a new technique: singular branches can be detected by asymmetries in the present distribution (taking mutation bias into account), then *reduced* revealing the original TMRCA.

We account for the tendency of bioprocesses to exaggerate the apparent mutation rate by using *reduction of singular lineages* to give a new phylogenetic method for computing mutation rates. This gives an average of .0031 mutations per generation, right between the meiosis results of Burgella and Navascue, Chandler. Indeed our theoretical phylogenetic rates are similar to actual laboratory meiosis rates, see Table 8. Despite our low mutation rates, “reduction of singular lineages” implies younger TMRCA. Also our low SD verifies the Kimura theory that rates are constant, at least for the human Y-Chromosome for 500 -

15,000ybp. However the large difference between rates, up to a factor of 100, implies that in the very long term that these may vary, probably as the geometry of the genome evolves. Furthermore we shall see that the essential equality of our rates with those from meiosis implies *a priori*, a generation length of 30 years. This agrees with Fenner(2005) who found a 30 year generation from field work among Hunter Gatherers. All of this provides theoretical validation.

However theoretically plausible the true test is accuracy, indeed our original motivation was to develop a Y-clock with accuracy approaching ^{14}C dating. This is verified by a comparison across a wide range of time, the first STR-clock giving consistent results for 500-15,000 ybp with accuracy about 11% (one SD). The usual sample error, uncertainty in mutation rates contributes to the total accuracy. However $\sim 1/2$ is inherent in the method itself, this being determined by large scale simulations of random trees with known TMRCA. It is easy to forget to include this, e.g. rarely computed for KAPZ methods. Comparison of 7, 15, 29 markers leads us to expect about 5% (one SD) for 120 markers.

We show our dates correlate with European history. In particular the dominant European Y-haplotype R1b1a2, expands from c4000BC, not reaching Anatolia before c3800BC. This contradicts previous clocks dating R1b1a2 to either the Neolithic Near East or Paleo-Europe. However our dates match R1b1a2 found in Yamnaya cemeteries of c3300BC, see Ramus Nielsen et al(2015), Svante Pääbo et al(2015). Our times and their places almost certainly implies R1b1a2 originates in the Russian Steppes, disproving the “out of Anatolia hypothesis” and instead giving strong evidence for the Indo-European hypothesis of Marija Gimbutas(1956). A different example is the dominant Amerindian Y-haplotype, Q1. Bortolini et al(2003) used 8 markers to give a TMRCA =11600BC (SD3000). Using recent data of Battaglia et al (2013) we find the TMRCA =13300BC with a 90% chance of occurring before the Canadian Ice Corridor opens, i.e. the initial migration was along the coast.

Very recently Batini et al (2015) computed TMRCA using the algorithm BEAST, with mutation rates for MYS sites on the Y-chromosome found from Icelandic genealogy by Helgason et al (2015). This is the complete opposite of our method, using $\sim 3 \times 10^6$ MSY sites compared with our 29 DYS, but samples sometimes as small as $N = 8$ while we have $N \sim 1000$. We compare all results in TABLE 10, see appendix. Their TMRCA are generally similar to ours, so together we support recent dates for R1b1a2 (and also verifying the Icelandic rates). For example Batini finds that R1b1a2 has TMRCA= 3550BC, compared with our 4000BC. But their sample of R1b1a2 (unlike ours) is M246* , i.e. major branches like L11, P312 are removed. So there is little branching and their results using BEAST and a KAPZ method RHIO are very similar. However when there is large scale branching our theory predicts their Bayesian method will give older results. This is seen for the SNP J2 where our TMRCA is about 5000 years younger. Our dates for J2 and G2a2 correlate with their expansion from the Neolithic Revolution. Another example is for J2b, which is presently concentrated in Eastern Europe. Now Nielsen et al, Reich et al used ADMIXTURE to find the initial Indo-European expansion into Central Europe during the Corded Ware Culture (3000-2000BC) was mainly of Siberian type DNA but with significant Middle Eastern component not found in the original Yamnaya Culture, probably from mixing in the Western Ukraine. This correlates with our finding for J2b, TMRCA =3286BC (cf. 10,000BC for Batini).

Figure 1 Singular Lineages in Random Trees

Singular Lineages

A fundamental problem is that present populations have highly overrepresented branches we call *singular lineages*. A well known example is the SNP L21 which is a branch of R1b1a2. Individuals identified as L21 are often excluded from R1b1a2 analysis because they skew the results. Such a singular lineage causes the variance to be much greater, even though the original *TMRCA* remains unchanged, see figure 1.

For Bayesian methods such lineages are very unlikely giving an even greater apparent *TMRCA*. However one cannot deal with singular branches by excluding them. For one thing, our method will show that 50% of markers show evidence of singular side branches, i.e. more than a SD from expected. For instance for a large sample of R1b1a2 we plotted the distribution about the mode for DYS markers GATA H4 and YCAI IIa. For the later there is little difference between the theoretical distribution (red) and the actual data (blue), however for GATA H4 one sees the large discrepancy (see SI 1 for all 33 DYS distributions, in Supplement). The ideal distribution is not symmetric because

Figure 2 Biased distributions

we have also accounted for bias in the mutation process, see Wakeley(1994). These singular lineages are very (mathematically) unlikely to arise from the stochastic system which is the mathematical basis of KAPZ (or the equivalent Monte-Carlo

process modeling Bayesian trees). We believe that the standard stochastic process is perturbed by “improbable” biological processes.

First, the Watson-Galton process introduced to explain the extinction of aristocratic lineages(see Kendall, 1975) implies lineages almost certainly die out. Conversely, natural selection causes some branches to flourish, e.g. the “kin selection” of W.D. Hamilton (1964), shows kin co-operation gives genetic advantages. Consider three examples with well developed DNA projects. Group A of the Hamiltons has approximately 100,000 descended from a Walter Fitzgilbert c1330AD. Group A of the Macdonalds has about 700,000 descendants from Somerfeld c1100AD, and Group A of the O’Niall has over 6 million descendants from Niall of the Seven Hostages, c300AD. These are elite groups with all the social advantages. One sees lines of chieftains, often polygamous. We emphasize kin selection because it seems dominant over natural selection for recent human lineages, certainly no one has claimed the O’Niall are genetically superior! Natural selection would cause similar branching over longer time scales. These singular lineages cannot be predicted by the stochastic model itself.

Reduction of Singular Lineages

Although our method is for general molecular clocks to be specific we focus on the Y-clock. Consider DNA Y-chromosome Segments (DYS) counting the “short tandem repeat” (STR) number of nucleotides. One uses these DYS micro-satellites, marked by $j = 1, \dots, n$, each individual i , $1 = 1, \dots, N$, has STR number $x_{i,j}$. The Y-chromosome is unchanged except for mutations $x_{i,j} \rightarrow x_{i,j} \pm 1$ occurring at rate μ_j per generation.

Modelling singular lineages requires a new stochastic system. The standard system is: at time t the probability P that the DYS markers have given STR vector X is governed by a homogenous system of discrete diffusion equations

$$\frac{\partial P}{\partial t} = \mu \Delta P$$

allowing variable mutation matrix $\mu = \mu(t, X)$. Our model will have constant mutation rate matrix μ in a discrete inhomogenous equation

$$\frac{\partial P}{\partial t} = \mu \Delta P + \delta$$

but with unknown sources $\delta = \delta(X, t)$ modeling unequal fertility. Solving either system for unknown μ or δ is an inversion problem. But inversion is unstable for such systems, also there is no unique solution.

Figure 3 Distribution of branching times

However in practise, up to a standard deviation, the inhomogenous system is a perturbation of the KAPZ system, $\delta(X, t) = 0$, so that “singular branches” are clear enough to be computed *a priori* without empirical considerations. These singular branches are then *reduced* revealing the original lineage.

We then compute a branching time t_j for each DYS marker j . Now the nonuniform branching process causes the t_j to be randomly distributed so their mean is not the TMRCA. Figure 3 for R1b1a2, shows the error spread for each t_j over 29 DYS sites (note the outliers!). One can combine the distributions for t_j give the “Branching Spectrum” (dotted curve). Now large errors and outliers means one cannot simply take the max t_j to be the TMRCA. Instead stochastic simulations of the branching process, using robust statistics to avoid outliers, find the most likely TMRCA. The effect of reduction is dramatic, e.g. the TMRCA for R1b1a2 changes from 7150BC(KAPZ) to 4000BC after singular reduction, using the same markers and mutation rates, see Figure 4 for effect on the Branching Spectrum.

Figure 4 Branching Spectrum before reduction(blue) and after.

In SI 6 we display similar for eight different SNP. Some of these are like a tight normal distribution showing that the expansion occurred quickly, e.g. L21. Others like J2 show the branching occurs over 100 generations.

Accurate Mutation rates

By relying on asymmetries of the distribution to find singular lineages we have to be aware that the mutation process itself might not be symmetric, this is called bias (Wakeley, 1994). If ignored we might be just detecting these asymmetries. Changing the model, the probability of a mutation per generation is

$$Pr[x_{i,j} \rightarrow x_{i,j} + 1] = \mu_{j,+1}, Pr[x_{i,j} \rightarrow x_{i,j} - 1] = \mu_{j,-1}.$$

If this marker is free from singular lineages we observe that the ratio of the frequencies to the left and right of the mode is

$$\frac{P_{j,1}(t)}{P_{j,-1}(t)} = \frac{\mu_{j,1}}{\mu_{j,-1}}.$$

which is time independent. So using eight very large SNP projects we find enough markers free of singularities to compute these ratios and their standard deviations, see Table 9 in Appendix and SI for computations. In particular about 33% of DYS markers show asymmetric ratios are significant, i.e. more than two SD from ratio 1. These asymmetric ratios play a very important role, for this ratio is all you need to detect a singular

lineage and reduce it. Of course not knowing the exact asymmetric ratio means that bootstrap methods are used extensively for singular reduction, both to compute values and SD.

These methods also imply a new way of computing mutation rates. Previously, there were methods based on meiosis data or phylogenetic studies of family DNA projects (which gave quite different rates). We begin with 8 very large SNP projects from FTDNA using 37 markers, of course with unknown TMRCA. We first reduce singular lineages. Then taking asymmetry into account we find mutation rates are the fixed points of an iterative process. This takes about 3 iterates to converge. These mutation rates are normally distributed with mean and SD. Discarding markers with mutation $SD > 33\%$ leaves us with 29 markers. We find this advanced phylogenetic method gives mutation rates close to those obtained from meiosis and nearly 1/2 the values obtained from the usual phylogenetic method. Further validation comes from finding that the equivalence of our rates with meiosis implies *a priori* a human generation of c30 years.

Results

Accuracy is verified by checking for consistency over all of European history beginning with Group A of three genealogical projects of medieval age:

Table 1: Medieval comparison

Group	TMRCA	[SD]	Oldest patriarch
Hamilton	1399AD	[180]	Fitzgilbert d1330AD
Macdonald	1060AD	[270]	Somerfeld c1100 – 1150AD
O’Niall	100AD	[260]	Niall 300AD/Conn100AD

Archeological finds convinced Marija Gimbutas(1956) to attribute Proto Indo-European (PIE) to the Yamnaya Culture c3000-4000BC of the Russian Steppes. This is consistent with mainstream linguistic theory, some even wrote of linguistic DNA, see Anthony (2007). But actual genetics was ignored because current genetic clocks for R1b1a2 pointed to the Renfrew Hypothesis(1987) that PIE spread from Neolithic Anatolia, c 6000BC. Or Mesolithic or Paleolithic, depending on the genetic clock. However no-one checked if their clock worked over the whole range of time for different lineages.

The next table shows the expansion times of the dominant European Y-haplotypes R1b1a2 & R1a1a. This data is from FTDNA projects for region X only using individuals with named ancestor from X. These independent results agree within the standard deviation, with dates matching the Corded Ware Culture, a semi-nomadic people with wagons and horses who expanded west from the Ukraine c3500BC. This is consistent with the oldest R1b1a2, R1a1a skeletons being from the Yamnaya Culture, c 3300BC, see R. Nielsen et al (2015) S. Pääbo et al (2015).

Table 2: Indo-European SNP

Region	R1b1a2[SD]	N	R1a1a [SD]	N
All	4004BC[700]	677	4012BC[720]	1270
Russia	NA		3417BC[780]	337
Poland	4030BC[1070]	65	4470BC[830]	876
Germany	3160BC[610]	438	3630BC[870]	190
Scandinavia	2700BC[690]	153	4740BC[1100]	140

The regional differences seem random suggesting rapid moment of whole peoples. Averaging overall (weighted by $1/\text{Variance}$ as the best linear estimate) gives R1b1a2 has TMRCA 4011BC(SD 580), R1a1a has TMRCA 3980BC(SD 380).

An interesting intermediate step occurs between the medieval and eneolithic. The mythical Irish Chronicles relate that the O’Niall descend directly from the first Gaelic High Kings, which tradition dated c1500 BC. The O’Niall have the unique mutation M222 which is a branch of the haplotype DF13 which is turn a branch of L21 and that of P312. The most likely TMRCA are

Table 3: Pre-Celtic SNP

SNP	TMRCA	SD
P312	2728BC	376
L21	2216BC	294
DF13	1867BC	328

These are dates for for early to late Bell Beaker Culture of Western Europe, 2800-1800BC. Indeed the only known Bell Beaker genome was found to be P312 with ^{14}C date 2300BC.

Table 4: Non FTDNA data

<i>data</i>	<i>Haplotype</i>	<i>N</i>	<i>TMRCA</i>	<i>[SD]</i>
<i>Underhill</i>	<i>R1a1a1</i>	974	3000BC	[800]
<i>Rootsi</i>	<i>G2a</i>	536	21000BC	[4200]

Our method requires large data sets and many markers which means we have to rely on data from FTDNA, finding 29 useable markers out of standard 37 they use. In fact many researchers have used FTDNA data, e.g Balaesque et al(2010). We think our method of reduction with robust statistics solves any problems with this data. To test this we used our method with R1a1a1 data obtained from Underhill(2015) with $N = 974$ and 15 useable markers, and G2a from Rootsi(2012) with a different 15 markers, see Table 4. With 29 markers the TMRCA for R1a1a1 is somewhat after our date for R1a1a, while G2a predates ours for G2a2(Table 5 are Western European only, see Table 7 for Anatolian).

Table 5 : TMRCA ybp for 29, 15, 7 markers

SNP	N	T29	SD	T15	SD	T7	SD
G2a2	1221	7654	1007	10800	2340	6580	2021
R1b1a2	660	6004	700	6800	1075	8264	2217
R1a1a	1270	6012	720	5270	840	5643	1665
I1	2898	4109	460	5114	1010	5799	1775
L21	1029	3927	415	4052	503	4252	882
U106	1533	4883	534	3043	710	4243	947
J2	1241	14490	2000	21060	3860	8814	2302
P312	971	4449	490	5340	748	5145	1128
Mean SD			12 %		18 %		26 %

Table 5 shows the results of extensive simulations using random subsets of our FTDNA data, for 29, 15(using Underhill markers) and 7 markers (Balaesque markers). In a fact our data for R1a1a for 15 markers is rather close to the Underhill data for R1a1a1.

However once you get down to 7 markers the confidence interval becomes too large for accuracy. Also it becomes difficult to deal with outliers. We might expect 5% accuracy with 120 markers.

An example with few markers is the R1b1a2 data of Balaesque et al(2010). Our method (this time with 7 useable markers) gave $SD \sim 30\%$. Now Balaesque used BATWING to suggest a Neolithic origin in Anatolia. With the same Cinnioglu(2004) data our method gives for Turkish R1b1a2 anything from the Ice Age to the Iron Age as seen in

Table 6: Balaesque Data

R1b1a2	N	TMRCA	SD
Eire	75	2046BC	1372
C.Europe	207	6383BC	2600
Turkey	69	5150BC	2700

Fortunately, once again, we find good data from FTDNA: the Armenian DNA project, see Table 7. By tradition the Armenians entered Anatolia from the Balkans c1000BC so they might not seem a good example of ancient Anatolian DNA. But some 100 generations of genetic diffusion has resulted in an Armenian distribution of Haplotypes J2, G2a2b, R1b1a2 closely matching that of all Anatolians, therefore representative of typical Anatolian DNA. We see that Anatolian R1b1a2 arrived after c3800BC, ruling out the Neolithic expansion c6000BC. When dealing with regional haplotypes, e.g. R1b1a2 in Anatolia, the TMRCA is only a upper bound for the arrival times, for the genetic spread may be carried by movements of whole peoples from some other region. This means one has to be careful interpreting regional data, e.g. the TMRCA for the R1b1a2(USA) is c4000BC but nobody thinks it arrived then.

Table 7: Anatolian SNP

<i>Armenian</i>	<i>N</i>	<i>TMRCA</i>	<i>[SD]</i>
<i>R1b1a2</i>	99	3780BC	[1000]
<i>G2a2b</i>	46	10000BC	[2500]
<i>J2</i>	97	12600BC	[2500]

Observe that our TMRCA for Armenian G2a2b (formerly G2a3) and J2 correlates with the first Neolithic farmers in the Middle East. From Table 5 we see J2, G2a2b for all of Western Europe (non-Armenian data). Our dates show J2 was expanding at the end of the Ice Age. Modern J2 is still concentrated in the fertile crescent, but also in disconnected regions across the Mediterranean. The old genetic model predicted a continuous wave of Neolithic farmers settling Europe (L.L. Cavalli-Sforza et al, 2004) . But you cannot have a continuous maritime settlement: it must be *leap-frog*. Also repeated resettlement from the Eastern Mediterranean has mixed ancient J2 populations, and our method gives the oldest date. On the other hand G2a2b shows exactly the dates expected from a continuous wave of Neolithic farmers across Central Europe. Our dates are consistent with recent findings that the majority of early Neolithic skeletons found in Western Europe are G2a2, c 5000BC, whereas the oldest R1b1a2 found in Western Europe is Bellbeaker c2300BC.

The problem of the peopling of the Americas has received as much attention. Y-chromosome analysis of (subarctic) Amerindians shows haplotype Q1 is dominant. Bortolini et al(2003) applied KAPZ with 8 markers to a sample of $N = 24$ to give a

TMRCAs for Q1 of c11600BC, (3000SD). More recent studies, e.g. Reich et al(2012), have been careful not to estimate a TMRCAs. We use 22 marker data from Battaglia et al(2013) who like Riech saw only one Q1 migration, but gave no dates. Ours are

Table 8: Battaglia Amerindian data

SNP	N	TMRCAs	SD
all Q1	437	12600BC	2000
Q1-L54	73	13300BC	2800
Q1-M3	335	10500BC	1770

This is supported by 29 marker FTDNA data from their New Mexico project:

Table 9: New Mexico FTDNA

SNP	N	TMRCAs	SD
NM Q1	110	16200BC	3000
NM M3	71	11030BC	2300

Thus we are 90% certain that the entry of Q1 into the Americas occurred before the Canadian Ice-Corridor was passable, c9500BC, see Mandryk, C.A.S. et al (2001). The Q1-M3 subgroup is found through most of the Americas, in fact is a majority of Amerindian Q1. Its later date is very interesting as it is about the time the Ice-Corridor opens. While it is probable that the subbranch L54 arrived with Q1 we are 90% certain that M3 did not arrive with the original migration. (We got 6000BC for the haplotype C3 which probably arrived with the later Na-Dene migration).

Discussion

Archeology, evolutionary biology, not to mention epidemiology, forensics and genealogy are just some of the applications of molecular clocks. Unfortunately current clocks have been found to give only "ballpark" estimates. Our method gives accurate time, at least for the human y-chromosome verified over the period 500 – 15,000ybp. Our main ideas hold for general molecular clocks, however our Y-clock has features specific to STR mutations. It would be important to develop reduction of singular lineages for other clocks.

Some geneticists thought natural selection makes mutation rates too variable to be useful. The problem is confusion between the actual biochemistry giving mutations and superimposed processes like kin selection producing apparently greater rates. Notice that the SD for our mutation rates is about 14%, previously many authors claimed about 10% yet had large differences with other studies.

Many applications to genetics, forensics, genealogy require the TMRCAs between just two individuals, or between two species, a classic method was given by Walsh (2001). While we are accurate for "big data", for this "two-body problem" one cannot determine what singular lineages the branching has been through. Just using our new asymmetric mutation rates will not work. So it would be important to find an accurate method.

Pääbo et al(2015), Nielsen et al (2015) observed all 6 skeletons from Yamnaya sites, c 3300BC by ¹⁴C dating, are either R1a1b2 and R1a1a. This involves very difficult genetic analysis of fossils which may not always be available. Also such analysis cannot date the origin of R1a1b2 and R1a1a. Our TMRCAs shows

both these haplotypes expanding at essentially the same time c4000BC. This and our later date for Anatolia, combined with Pääbo et al, implies that R1b1a2 and R1a1a must have originated in the Yamnaya Culture.

In checking accuracy we ran into the question of the origins of Indo-European. Although there are genes for language there is certainly none specific to Indo-European language. Thus inferences have to be indirect. Marija Gimbutas(1956) saw patterns in symbolism and burial rituals suggesting the Yamnaya Culture was the cradle of Indo-European. Also their physiology was robustly Europeanoid unlike the gracile skeletons of Neolithic Europe, but this could be nutrition and not genetic. From the above we conclude that the spread of this robust type into Western Europe in the late Neolithic marked an influx of Steppe nomads. Now if R1b1a2 had been shown to spread from Anatolia c6000BC it would have been taken as strong evidence for "out of Anatolia" because of the association of R1b1a2, R1a1 with Indo-European languages. But our accuracy check showed that it was G2, J2 that spread with the Neolithic Expansion from Anatolia. Now these have been associated with Caucasian languages or Semitic, but never with Indo-European.

Methods

This work is biomathematical theory validated by data from published sources. We give full mathematical details in the following appendix. Also see Supplementary Information (SI) for detailed algorithms and MATHEMATICA worksheets, examples and some data. To verify the theory and compute mutation rates we use diverse data, from FTDNA y-haplotype projects for G2a2b, R1b1a2, R1a, I1, L21, U106, J2, P312. Also we used regional projects for Germany, Scandinavia, Poland and Russia for their R1b1a2, R1a1a data. The Armenian DNA project was important for its R1b1a2, J2 and G2a2b data. We also used DNA projects M222 (O'Niall), Macdonald (Group A which is R1a1a), Hamilton (group A which is I1). This was compared with non FTDNA data from Balaesque, Underhill, Rootsi and Battaglia et al.

Literature cited

- Ammerman, A. J. & L.L. Cavalli-Sforza (1984) *Neolithic transition and the genetics of populations in Europe* Princeton University Press.
- Anthony, D. W. (2007) *The Horse, the Wheel and Language*, Princeton Univ. Press, Princeton
- Ayala, F. J. (1997) , Vagaries of the molecular clock, *Proc Natl Acad Sci USA* 94:7776-7783.
- Ayala, F. J. et al (2001), Erratic overdispersion of three molecular clocks: GPDH, SOD, and XDH , *Proc Natl Acad Sci USA* vol. 98 no. 20 pp 11405-11410
- Balaresque, P. et al (2010), A predominantly Neolithic Origin for European paternal Lineages, *PLoS Biol* ; 8: e1000285.
- Batini C. et al. (2015) , Large-scale recent expansion of European patrilineages shown by population resequencing. *Nature Communications* 6: 7152 doi:10.1038
- Battaglia, V. et al. (2013) , The First Peopling of South America: New Evidence from Y-Chromosome Haplogroup Q, *PLoS ONE* 8(8): e71390. doi:10.1371/journal.pone.0071390
- Beaumont, M. A. et al (2002). Approximate Bayesian computation in population genetics *Genetics* 162:2025-2035
- Bortolini, M.C. et al, (2003), Y-Chromosome Evidence for Differing Ancient Demographic Histories in the Americas, *Am. J. Hum. Genet.* 73:524-539
- Burgarella, C. and M. Navascue (2011) , Mutation rate estimates for 110 Y-chromosome STRs combining population and father-son pair data, *European Journal of Human Genetics* 19:70- 75; doi:10.1038/ejhg.2010.154
- Busby, G. (2012), The peopling of Europe and the cautionary tale of Y chromosome lineage R-M269, *Philos Trans R Soc Lond. Biol Sci.*; 279(1730):884-92. doi: 10.1098/rspb.2011.1044
- John F. Chandler, J.F. (2006), Estimating Per-Locus Mutation Rates, *Journal of Genetic Genealogy* 2:27-33
- Cavalli-Sforza, L.L. et al (2009), Y chromosome diversity, human expansion, drift, and cultural evolution, *Proc Natl Acad Sci USA* 106:20174-20179.
- Cinnioglu, C. et al. (2004) , Excavating Y-chromosome haplotype strata in Anatolia, *Hum Genet.* 114:127- 148.
- Drummond, A. J. , A. Rambaut (2007), BEAST: Bayesian evolutionary analysis by sampling trees, *BMC Evol. Biol.* 7, 214
- Drummond, A. J et al (2005), Bayesian coalescent inference of past population dynamics from molecular sequences, *Mol. Biol. Evol.* 22: 1185-1192
- Durrett, R. (2002) *Probability Models for DNA Sequence Evolution* Springer, New York
- Edmonds, C. , A.Lillie, L.L.Cavalli-Sforza, (2004), Mutations arising in the wave front of an expanding population, *Proc Natl Acad Sci USA* 10: 975-979.
- Elango, N. , J.W. Thomas , S. V. Yi (2006), Variable molecular clocks in hominoids, *Proc Natl Acad Sci USA* vol. 103, no.5,1370-1375
- Epstein, C. and R. Mazzeo (2013) *Degenerate Diffusion Operators Arising from Population Biology*, Princeton
- Fenner, J.N. (2005) , Cross-Cultural Estimation of the Human Generation Interval for Use in Genetics-Based Population Divergence Studies, *American Journal of Physical Anthropology* 128, 415-428.
- Fu, Y. X. , W.H. Li (1993), Statistical tests of neutrality of mutations, *Genetics* 133: 693-709
- Gimbutas, M., (1956) *The Prehistory of Eastern Europe, Part 1*, Cambridge: American School of Prehistoric Research #20.
- Hamilton, W.D. (1964), The genetical evolution of social behaviour. I, II, *Journal of Theoretical Biology* 7 (1) pp 1- 52.
- Helgason, A. et al. (2015), The Y-chromosome point mutation rate in humans., *Nat. Genet.* doi:10.1038/ng.3171
- Ho, S.Y., M.J. Phillips, A. Cooper, A.J. Drummond (2005), Time dependency of molecular rate estimates and systematic overestimation of recent divergence times, *Molecular Biology & Evolution* 22 (7): pp 1561-1568.
- Hoffecker, J.F. and S.A. Elias (2007), *Human ecology of Beringia*, Columbia University Press, New York.
- Lynch, M. (2010), Rate, molecular spectrum, and consequences of human mutation, *Proc Natl Acad Sci USA* vol. 107 , no. 3 :961-968
- Kendall, D.G. (1975), The Genealogy of Genealogy Branching Processes before (and after) 1873, *Bulletin of the London Mathematical Society* 7 (3) pp 225- 253.
- Kimura, M. (1968) , Evolutionary rate at the molecular level, *Nature Vol.* 217, 624 - 626 , doi:10.1038/217624a0
- Kimura M, and T.Ohta (1973), Mutation and evolution at the molecular level. *Genetics* 73(Suppl):19 - 35
- Knoll, A.H. et al, (2011) , Estimating the timing of early eukaryotic diversification with multigene molecular clocks , *Proc Natl Acad Sci USA* vol. 108, no. 33: 13624-13629
- Korber, B., et al (2000) Timing the ancestor of the HIV-1 pandemic strains. *Science*, 288: 1789 -1796.
- Lacana, M. et al (2011), Ancient DNA suggests the leading role played by men in the Neolithic dissemination, *Proc Natl Acad Sci USA* vol. 108, no.45, 18255-18259
- Mandryk, C.A.S. et al (2001), Late Quaternary paleoenvironments of Northwestern North America: implications for inland versus coastal migration routes, *Quaternary Science Reviews* 20: 301- 314.
- Menozzi, P. , A. Piazza, L.L. Cavalli-Sforza (1978), Synthetic maps of human gene frequencies in Europeans, *Science* 201:786-792
- Myre, N. et al (2011), A major Y-chromosome haplogroup R1b Holocene era founder effect in Central and Western Europe, *European Journal of Human Genetics* 19(1) 95-101
- Nei, M. and S.Kumar (2000) *Molecular Evolution and Phylogenetics*. Oxford University Press, New York (333 pp).
- Nielsen, R. and J. Wakeley (2001), Distinguishing migration from isolation: a Markov chain Monte Carlo approach, *Genetics* 158:885-896
- Nielsen, R. (Editor) (2005) *Statistical Methods in Molecular Evolution*, Springer.
- Olver, F.W.J. (1964) *Bessel functions of integer order* 355- 434 *Handbook of Mathematical Functions* edited by M. Abramowitz National Bureau of Standards, Washington.
- Ohta, T. (2002), Near-neutrality in evolution of genes and gene regulation. *Proceedings of the National Academy of Sciences* 99 (25): 16134-16137
- Pääbo, S. et al (2014), Ancient human genomes suggest three ancestral populations for present-day Europeans, *Nature* 513, pp 409-413.
- Reich, D. et al (2012), Reconstructing Native American population history, *Nature* 488:370-375
- Reich, D. et al (2015) , Massive migration from the steppe was a source for Indo-European languages in Europe, *Nature* 14317, doi:10.1038/nature14317
- Nielsen, R. et al (2015), Population genomics of Bronze Age Eurasia *Nature* 522:167-172 doi:10.1038/nature14507
- Renfrew, A. C. (1987) *Archaeology and Language: The Puzzle of*

- Indo-European Origins*, London:Pimlico.
- Rootsi S. et al (2012), Distinguishing the co-ancestries of haplogroup G Y-chromosomes in the populations of Europe and the Caucasus, *Eur J Hum Genet.* 20(12):1275-82. doi: 10.1038/ejhg.2012.86.
- Underhill, P.A. et al, (2015), The phylogenetic and geographic structure of Y-chromosome haplogroup R1a, *European Journal of Human Genetics* 23:124-131; doi:10.1038/ejhg.2014.50
- Wakeley, J. (1994), Substitution-rate variation among sites and the estimation of transition bias, *Mol. Biol. Evol.* 11:436-442.
- Wakeley, J. (2008) *Coalescent Theory: An Introduction* Roberts & Company Publishers, Greenwood Village, Colorado.
- Wakeley, J. (2013), Coalescent theory has many new branches, *Theoret. Pop. Biol.* 87:1-4.
- Walsh, B. (2001), Estimating the Time to the Most Recent Common Ancestor for the Y chromosome or Mitochondrial DNA for a Pair of Individuals, *Genetics* 158: 897-912
- Wehrhahn, C. F.(1975), The evolution of selectively similar electrophoretically detectable alleles in finite natural populations, *Genetics* 80: 375- 394.
- I. J. Wilson, M.E. Weale, D.J. Balding (2003), Inferences from DNA data: population histories, evolutionary processes and forensic match probabilities, *J Roy Statist Soc A.* (2003);166:1-33.
- Zhivotovsky, L.A. (2000), Estimating Divergence Time with the Use of Microsatellite Genetic Distances, *Molecular Biology and Evolution* 18, No 5:705-709
- Zuckerandl, E. and L. Pauling (1965) *Evolutionary divergence and convergence in proteins.* In Bryson, V.and Vogel, H.J. (editors). *Evolving Genes and Proteins.* Academic Press, New York. pp. 97-166.
- Zuckerandl, E. and L. Pauling, L.B. (1962), Molecular disease, evolution, and genic heterogeneity, in Kasha, M. and Pullman, B (editors). *Horizons in Biochemistry.* Academic Press, New York. pp. 189-225.

Appendix

Biomathematical theory

Our method requires a return to basic principles.

Fundamental Solutions

The Y-chromosome has DYS marked by $j = 1, \dots, n$, where one can count the STR number x_j . Consider the probability $P_{j,k}$ (at time t generations) that at marker j we have $x_j = k$. This satisfies the homogenous stochastic system

$$\frac{P_{j,k}(t)}{dt} = -\mu_j P_{j,k} + \sum_{m>0} \mu_{j,-m} P_{j,k-m} + \mu_{j,m} P_{j,k+m}$$

This homogenous system gives a uniform expansion from a single patriarch.

The system is essentially the model of Wehrhahn(1975) who had $\mu_{j,-1} = \mu_{j,1}$. We introduce asymmetric mutations with total rate

$$\mu_j = \sum_{m>0} \mu_{j,-m} + \mu_{j,m}$$

About 50% of DYS markers show asymmetric mutations, i.e. $\mu_{j,-1} \neq \mu_{j,1}$.

We use the generator function with complex variable z

$$G(z, t) = \sum_{k=0}^{\infty} P_{j,k} z^k,$$

(see Durrett, 202), and normalized initial condition $x_j = 0$ or $P_{j,0}(0) = 1$:

$$G(z, t) = \exp[-\mu_j t + t \sum_{m>0} \mu_{j,-m} z^m + \mu_{j,m} z^{-m}]$$

Then G can be expanded in powers of z to give $P_{j,k}(t)$. Now for the simplest asymmetric case, with only one step mutations, we have $G(z, t) = e^{-\mu_j t} e^{t\mu_{j,-1}z} e^{t\mu_{j,1}/z} =$

$$e^{-\mu_j t} \left\{ \sum_{m=0}^{\infty} \frac{\mu_{j,-1}^m}{m!} (zt)^m \right\} \left\{ \sum_{m=0}^{\infty} \frac{\mu_{j,1}^m}{m!} (t/z)^m \right\}$$

so using the Hyperbolic Bessel Function of Order $k \geq 0$, see Olver (1964)

$$I_k[u] = \sum_{m=0}^{\infty} \frac{u^{2m+k}}{2^{2m+k} m! (m+k)!},$$

we see that the homogenous system has fundamental solution

$$P_{j,k}(t) = e^{-\mu_j t} \left(\frac{\mu_{j,1}}{\mu_{j,-1}} \right)^{k/2} I_{|k|}[2t\sqrt{\mu_{j,-1}\mu_{j,1}}]$$

From the fundamental solution we find, independently of time

$$\frac{P_{j,1}(t)}{P_{j,-1}(t)} = \frac{\mu_{j,1}}{\mu_{j,-1}},$$

which we call the *asymmetric ratio*. It will be repeatedly used.

Of course the actual initial value is not $x_j = 0$ but was usually taken to be the mode m_j which was assumed to be the value for original patriarch. From the present distribution of data we use the frequency

$$f(j, k) = \frac{\text{Count}(x_j(i) = k)}{n}.$$

One problem with the KAPZ formula is that higher frequencies $f(j, k)$, $|k| = 2, 3, \dots$ are overrepresented in the actual data. This is because the probability of a spontaneous two step mutation is much higher than the product of two one step mutations. So instead we use the frequency to solve the transcendental equation for the unknown t

$$f(j, 0) \sim P_{j,0}(t) = e^{-\mu_j t} I_0[2t\sqrt{\mu_{j,-1}\mu_{j,1}}]$$

This nonlinear equation is easily solved via mathematical software such as MATHEMATICA (I used version 9 running on a boosted 2014 iMac which has accurate hyperbolic Bessel functions. Earlier versions on older iMacs gave inaccuracies so one had to compile one's own functions). Using this formula resolves some other problems with the KAPZ method even for a uniform flow, e.g. $\mu_{j,-1} \neq \mu_{j,1}$ gives an extra quadratic term which if ignored causes large errors.

Heterogeneous diffusion equation

However the main problem is singularities in the stochastic process. For a uniform stochastic process, $1 - P_{j,0}(t) \sim 1 - f(j, 0)$ is the probability of some mutation. So the expected variance is $f(j, 0)(1 - f(j, 0))$. Thus if the actual data variance $V_j \gg f(j, 0)(1 - f(j, 0))$ we are not uniform. Now a sublineage of very high fertility increases variance, giving apparently greater TMRCAs although it is unchanged. One finds similar results for Bayesian methods.

The correct approach to nonuniformity assumes at times t_i (generations ago) a certain proportion $0 \leq \rho_i \leq 1$ of the present population originated from a "virtual patriarch" with an initial STR value m_i . The resulting system :

$$\frac{dp_{j,k}(t)}{dt} = -\mu_j p_{j,k} + \sum_{m>0} \mu_{j,-m} p_{j,k-m} + \mu_{j,m} p_{j,k+m} + d\rho$$

i.e. $d\rho$ are atoms of weight ρ_i with STR value m_i occurring at time t_i . As the system is linear and isotropic the solution is a combination of fundamental solutions P of the homogenous system. Thus the present distribution $f(j, k)$ is

$$f(j, k) \sim p_{j,k}(t) = \sum_i \rho_i P_{j,k-m_i}(t_i)$$

This allows us to consider populations mixed by having singular lineages from overfertile patriarchs, or by actual immigration from the outside. For example Nielsen, R. and J. Wakeley (2001) considered mixed populations with two sources. However in our general case has no obvious limit on the number of sources. The inverse problem seeks to find singularities from present data. Unfortunately, in general, inversion is ill posed for such systems like the heat equation. This instability produces poor accuracy. Furthermore there is no unique solution, e.g. the present distribution could have been created yesterday.

However we find that $\sim 50\%$ of the DYS markers show no significant difference from the uniform expansion of a single patriarch, i.e. the data variance V_j is close to the expected variance $f(j, 0)(1 - f(j, 0))$. The other markers show at most one significant side branch, i.e. there is an original branch starting at time $t_{j,0}$ with STR m_0 and a second one with STR $m_1 = m_0 \pm 1$ at time $t_{j,1} < t_{j,0}$ with significant $0 < \rho_1 < \rho_0$. Thus we have mane able 1-2 sources per marker.

Reduction of Singular Lineages

We locate these singular lineages by looking for asymmetries in the distribution. For a uniform flow from a single patriarch the frequency of STR value k is given by $f(j, k) \sim P_{j,k}(t)$. The asymmetric ratio:

$$\frac{f(j, 1)}{f(j, -1)} \sim \frac{P_{j,1}(t)}{P_{j,-1}(t)} = \frac{\mu_{j,1}}{\mu_{j,-1}},$$

is completely independent of time t . Therefore if say

$$\frac{f(j, 1)}{f(j, -1)} >> \frac{\mu_{j,1}}{\mu_{j,-1}},$$

we have a singular lineage at $k = +1$. Thus the excess at $k = +1$ is

$$f(j, +1) - f(j, -1) \frac{\mu_{j,1}}{\mu_{j,-1}}$$

To first order approximation then frequency $f(j, +2)$ is due to this singularity at $j = +1$ which therefore gave a contribution

$$f(j, +2) \frac{\mu_{j,-1}}{\mu_{j,+1}}$$

to $k = 0$. Thus removing the effect of the singularity at $k = +1$ leads to new frequencies

$$f^*(j, -1) = f(j, -1)$$

$$f^*(j, 0) = f(j, 0) - f(j, +2) \frac{\mu_{j,-1}}{\mu_{j,+1}}$$

$$f^*(j, +1) = f(j, +1) - f(j, -1) \frac{\mu_{j,1}}{\mu_{j,-1}}$$

These of course are no longer normalized so we rescale to obtain the renormalized frequency $F(j, k)$, e.g.

$$F(j, 0) = \frac{f^*(j, 0)}{f^*(j, 0) + f^*(j, -1) + f^*(j, +1)}$$

which will be used to compute the expansion time for marker j . There are similar formulae if the singularity was at $k = -1$.

However there is sampling error both in the frequencies and the $\mu_{j,1}, \mu_{j,-1}$. So we bootstrap taking into account these uncertainties, running the computation thousands of times. Generally we find the branch singularity is always one of $k = 0, +1, -1$ with no SD. In a few cases the singularity may seem to wander between $k = 0, +1, -1$. So in the case of a wandering singularity we obtain a distribution over $k = 0, +1, -1$ with a mean and SD. In these cases we find the singularity is relatively small and does not make much difference to the final result. However to have a stable method we do not throw out these wandering singularities but in the algorithm use the mean to average between $k = 0$ and $k = \pm 1$, e.g. if the mean is $k = 0$ then we use the original unreduced frequency.

Notice that we assume at most one side branch. In theory there could be many and solving for these produce even better approximations to the present data. In fact you could get perfect matching but find the atoms were created yesterday! The thing is that while many markers show significant deviation from a uniform flow from a single patriarch, after we have carried out reduction for one possible side branch we find no significant difference from a uniform flow, i.e. the difference is within the

SD. This is of course an approximation, the next level beyond Zuckerkandl and Pauling, but given the noise in the data perhaps the best we can do. Later we further reduce the effect of outliers by using robust statistics.

Reducing the singular lineages increases the frequency $f(j, 0)$ of the mode and decreases the computed $TMRCAs$. But as the method of reducing singularities does not respect higher frequencies $f(j, k)$ it follows the KAPZ formula cannot be used and instead we use the probability of no mutations, i.e. solve

$$F(j, 0) = e^{-\mu_j t} I_0[2t \sqrt{\mu_{j,-1} \mu_{j,1}}]$$

This is done for each DYS marker j , giving expansion times $t_1, \dots, t_N$ for each marker, with computed CI. (An extra fixed source of error is the uncertainty in the mutation rates which we deal with later). We find the reduction of singularities makes striking difference to the t_j of the effected markers, often a reduction of $\sim 50\%$ for $TMRCAs$.

Now the existence of side branches implies that the main branch could itself have been the side branch for an earlier branch that did not survive. Thus we do not expect the expansion times $t_1, \dots, t_n$ for each marker to be essentially equal, i.e. they are not within the SD of each other. Indeed we see that the distribution of the times t_j for different markers are almost certainly not randomly arranged about a single $TMRCAs$ T but distributed from T to the present. This is seen whether you use reduction or not, or our mutation rates or not. (For a given population one could scale mutation rates to get equal t_j , but then applying these adhoc mutation rates to other populations does not yield the same values). The spread out distribution of surviving branches is another verification of our theory of many extinctions, few survivors. The distribution of the times t_j for different markers we call the branching distribution, which is now discussed.

The Branching Distribution

The times t_j for different markers are sorted from the youngest to the oldest, forming a sequence $t_1^*, \dots, t_n^*$. The generation of these branches is by an unknown probability distribution $d\tau_0$ over $[0, T]$. We model $d\tau_0$ by assuming a surviving lineage is generated at random with probability $\beta \Delta t$ in time period $[t, t + \Delta t]$, multiplied by the probability that the branching hasn't already occurred. The constant β averages fertility and extinction rates, the chance of a new lineage surviving. As $\beta \rightarrow \infty$ we get current theory where all lineages originate from a single patriarch at time T . Simulations with the data show that β varies in the range 1 to ∞ . We make no a priori estimate of β , unlike Bayesian methods where an overall fertility rate is a predetermined parameter. Instead our stochastic simulation will find the most likely β, T in each case. Assuming independence, then the generation of branches follows the well known exponential distribution:

$$\tau_0[t] = \text{Exp}[\beta(t - T)] \text{UnitStep}[T - t]$$

Notice this implies a finite probability that some markers have essentially zero mutations. This is actually seen in examples. Both the Hamilton Gp A and Macdonald Gp A have number of individuals $N > 100$. For the time scale of > 700 years we do not expect there is more than one marker out of 33 which shows absolutely no mutations from the mode. In fact in both cases there are 8 markers where all N individuals have exactly the same STR value.

We also tested this theory in simulations where we obtained random branching times based on the density for various values

of T, β . In SI 2 we show results for "pseudo R1b1a2", i.e. with $TMRCAs$ $T = 200$ generations. This looks like the actual spectrum for R1b1a2 for $1/\beta \sim 20$ generations, i.e. a lop sided bell curve showing fast expansion. For $1/\beta \sim 100$ the curve looks like our curve for J2, i.e. spread out indicating much branching.

Estimating the parameter T for an exponential distribution is a well known problem of statistics. Kendall proved the best estimate for T would be $\max t_j$. Unfortunately there is also considerable error $\lambda_j\%$ for the mutation rates μ_j . Later we give a method for reducing this error, even so we find the SD in the range 10% – 30% which gives corresponding range in error for each t_j . We understand that the t_j are being generated by the distribution $d\tau_0$ but superimposed on this is a further uncertainty due to mutation rates etc. In particular the largest t_j may be wildly inaccurate. Also we found that simply taking the average consistently underestimates the $TMRCAs$ by a wide margin.

Assuming the mutation rates have normal distribution with mean μ_j and variance $\lambda_j^2 \mu_j^2$, the t_j have SD $t_j \lambda_j$. Thus the actual data for t_j^* has probability density function for $s > 0$

$$d\tau(s) = \int_0^T \frac{e^{(t-T)/\beta}}{\beta} \frac{e^{-\frac{(t-s)^2}{2v}}}{\sqrt{2\pi v}} dt.$$

The variance v depends on two sources. First from the uncertainty in mutation rates, for each of the n markers we get variance λ_j^2 , giving total

$$v_1 = \frac{1}{n} \sum_j \lambda_j^2$$

However a small sample also has inherent error from sampling. We are measuring the probability that there is a mutation. This is binomial with probability $H_j(t)$ given by

$$H_j = 1 - P_{j,0}(t) = 1 - e^{-\mu_j t} \left(\frac{\mu_{j,1}}{\mu_{j,-1}} \right)^{k/2} I_0[2t \sqrt{\mu_{j,-1} \mu_{j,1}}]$$

Hence for sample size N there is variance $H_j(1 - H_j)/N$, so the variance in time due to this is scaled by the derivative giving:

$$v_2 = \frac{H_j(1 - H_j)}{N(H'_j)^2}$$

The function H'_j has actually to be computed as an inverse function depending on H_j . Therefore the total variance averaged over all n markers is $v = v_1 + v_2$. Although for large samples ($N > 1000$) the second term is insignificant it does effect the results once you get to $N = 100$. In our algorithm the branching distribution is used to generate large numbers of random branching times so as to bootstrap error estimates. It turns out much faster to compile the distribution function as a table which can be repeatedly called on.

Estimating TMRCAs by Robust Statistics

Inaccurate large values of t_j^* are mitigated by using "robust" statistics with quintiles instead of means/variances. Using FTDNA data we began with 37 markers. However the 4 markers of DYS464 are unordered and cannot be used. Also we find that markers DYS 19/394, 385b, 459b, CDYb have errors $> 33\%$ in mutation rates so are not used. (These are some of the most popular ones in the literature!). We find if we just use the 15 markers with errors $< 15\%$ the final SD is about 13% compared

with 12% using 29. So usually we have $n = 29$ markers and take “quintiles” $\theta^* = (t_9^*, t_{12}^*, t_{15}^*, t_{18}^*, t_{21}^*)$. This means that tail end data is not discarded but kept as the information there are 8 values of $t_j^* > t_{21}^*$, which effectively deals with outliers. Bootstrap methods give the confidence interval CI for each quintile.

Thus we wish to find the best estimate of T given θ^* (and CI). This well known statistical problem was investigated by Stochastic Simulations (SS). We also tried Maximum Likelihood Methods which gave similar results but with larger CI. Monte-Carlo Methods are used to produce very large numbers ($\sim 10^7$) of T , β with corresponding Distribution. These randomly generate ordered times ($s_1 \dots s_{29}$) for which we take the quintiles $\theta = (s_9, s_{12}, s_{15}, s_{18}, s_{21})$. We filter by requiring that θ close to the data θ^* , i.e. $\|\theta^* - \theta\| < rSD$. We tried $r = 3, 2, 1, 0.5$ obtaining almost identical final results, subsequently use $r = 2$. This gives a stochastic neighborhood $\mathcal{U}$ of θ^* typically containing $> 10^5$ sets of data but with T is known for each $\theta \in \mathcal{U}$. Thus we can construct a quasilinear estimator:

$$QL(s_9, s_{12}, s_{15}, s_{18}, s_{21}) = q_1 s_9 + q_2 s_{12} + q_3 s_{15} + q_4 s_{18} + q_5 s_{21},$$

and use least squares over $\mathcal{U}$ to find constants (q_1, q_2, q_3, q_4, q_5) minimizing

$$\|q_1 s_9 + q_2 s_{12} + q_3 s_{15} + q_4 s_{18} + q_5 s_{21} - T\|.$$

The (q_1, q_2, q_3, q_4, q_5) are computed in MATHEMATICA We test this by applying the QL to all of $\mathcal{U}$, unsurprisingly

$$Mean_{\mathcal{U}}[q_1 s_9 + q_2 s_{12} + q_3 s_{15} + q_4 s_{18} + q_5 s_{21} - T] \sim 0$$

What is important is that we find the uncertainty in the SS itself. Actually this depends on the data and is calculated in each case but for our examples we find

$$SD_{\mathcal{U}}[q_1 s_9 + q_2 s_{12} + q_3 s_{15} + q_4 s_{18} + q_5 s_{21} - T] \sim .05 T$$

Finally the quasilinear estimator is applied to the experimental data

$$(t_9^*, t_{12}^*, t_{15}^*, t_{18}^*, t_{21}^*)$$

to obtain our best estimate of T . Application of QL computes the SD for our data, giving part of the overall SD. This must be combined with the SD coming from the uncertainty in the SS. Overall we find that our method has SD $\sim 12\%$, this includes variances from our data, mutation rates and uncertainty in the SS. Also we tested with 7, 13 “quintiles” instead of 5 obtaining almost identical TMRCA and a slight improvement in accuracy to $< 11\%$. We also tested with 15 and 7 markers. Here one must use “quintiles” $\tau = (t_5^*, t_8^*, t_{11}^*)$, $\tau = (t_3^*, t_5^*)$, respectively with all the loss of accuracy that implies. See Table 5 for comparisons using 29, 15, 7 markers on same data.

Accurate Mutation rates

Any genetic clock depends on reasonably accurate mutation rates. The meiosis method looks for mutations in father-son studies. However typical rates of $\mu = .002$ would require nearly 50,000 pairs to get an SD of 10%. The phylogenetic approach studies well developed DNA/genealogy data. So inverting the KAPZ formula would yield accurate rates. However, *singular lineages* makes this problematic.

To compute our rates we apply our theory to the large DNA projects for the SNP M222, L21, P312, U106, R1b1a2, I1, R1a1a. This avoids dealing with populations such as family DNA projects which are self selecting, i.e only those with the correct

surname which neglects distant branches. Also we have very large samples, our average $N > 1000$. Greater accuracy should come from more generations and individuals. The problem is that we do not know their TMRCA.

Asymmetric Mutation

However before computing mutation rates we must consider asymmetric mutations, i.e. the left and right mutation rates $\mu_{j,-1} \neq \mu_{j,1}$. For a uniform stochastic process we again use the asymmetric ratio

$$\frac{p_{j,1}(t)}{p_{j,-1}(t)} = \frac{\mu_{j,1}}{\mu_{j,-1}} = \frac{A_j}{1 - A_j}$$

to define the *asymmetric constant* $A_j \in [0, 1]$ for marker j . For example $A_j = 0.5$ is complete symmetry. Of course singularities will effect this ratio, however these only occur $< 50\%$ of markers. Thus for each marker, SNP we compute this ratio. We find the SD for each SNP is relatively small while the difference between SNP can be large. However for each marker, using 8 SNP enables outliers to be easily removed leaving allowing us to use simple linear regression: i.e. average of the A_j over the remaining SNP groups. We see that asymmetry is a real effect: 50% of the A_j are more than two SD from symmetry $A_j = 0.5$.

Observe this is significant. The total second moment is

$$\sum_j \sum_{k=-\infty}^{k=\infty} k^2 P_{j,k} = t \sum_j \mu_j + t^2 \sum_j (\mu_{j,1} - \mu_{j,-1})^2$$

So using all our 33 DYS markers with our μ_j , we compute constants

$$\mu = \sum_j \mu_j = .12006, \tau = \sum_j (\mu_{j,1} - \mu_{j,-1})^2 = 0.000236$$

The classical KAPZ formula gives variance $V = \mu t$ compared to the corrected formula $\mu t + \tau t^2$. The uncorrected KAPZ gives an overestimate $> 400\%$ for > 200 generations. This effect can be nullified by using the mean instead of the mode, variance instead of the second moment, however failing to do so gives a large error. Furthermore other methods which assume symmetric mutations will also be inaccurate. Having estimates on the asymmetry is essential to our method because we find singular lineages by looking for asymmetry in the data. Any such anomaly needs to be significantly greater than the natural asymmetry.

Mutation Rates as a fixed Point

Next we compute mutation rates using 8 very large SNP groups. First, using the asymmetric constants we find singular lineages and reduce their effect. We take account of the error in the A_j by a bootstrap technique, which gives the variance for each frequency $f(j, 0)$.

For each SNP k if DYS markers j started their mutations at the same time TMRCA T_k we could calculate mutation rates μ_j via

$$(1) \quad f(j, 0) = e^{-\mu_j T_j} I_0[2T_j \sqrt{\mu_{j,-1} \mu_{j,1}}],$$

or rather average the 8 different μ_j we would obtain. However because of branching caused by extinction of lineages the different markers do not originate at the same time but at different times t_j . In any case we don't know the TMRCA T_k .

Instead we use the log modal, i.e. defining T_k^* to be the log mean over the middle set of t_j . So for a fixed marker j the

normalized times $\tau_{k,j} = t_j - T_k^*$, $k = 1, 2, \dots, 8$ should straddle 0, i.e have a 50% chance of being positive. However the wrong choice of μ_j would give a bias, e.g. if μ_j is too big then the t_j are too small, so the $\tau_{k,j}$ will tend to be negative. In fact this is what we see if the mutation rates $\mu_j = .002$ were chosen. In the SI figure shows the $\tau_{k,j}$, $k = 1, \dots, 8$ bunched around a nonzero point. Thus we try to find μ_j so that the $\tau_{k,j}$, $k = 1, 2, \dots, 8$ straddle zero. However the $\tau_{k,j}$, $k = 1, 2, \dots, 8$ depend nonlinearly on the rates μ_j , as does the log modal T_k^* , $k = 1, \dots, 8$. This nonlinear regression problem is solved by an iterative scheme which starts with any reasonable set of mutation rates and corrects them. We find any reasonable first choice iterates to the same final answer. So choose $\mu_j = .002$ to begin. Suppose at some stage we have apparent mutation rates μ_j . Then, for each SNP, and each marker we solve equation (1) to obtain the apparent t_j . For each SNP $k = 1, \dots, 8$ we compute the mean log modal T_k^* . At the next we get new rates μ_j^* by solving the transcendental equation

$$f(j, 0) = e^{-\mu_j^* T_k^*} I_0[2T_k^* \sqrt{\mu_{j,-1}^* \mu_{j,1}^*}]$$

Averaging μ_j^* , $k = 1, \dots, 8$ we get our next set of μ_j of mutation rates. This method would be effected by a marker showing a singular lineage. Fortunately these are few in number and by comparison between the different SNP we remove the outliers. We then repeat the process, computing T_k^* again with the new rates, and another set of mutation rates.

One problem is that the iterates could tend to decrease to zero or increase to ∞ , as we are only calculating relative rates. To prevent this we renormalize after each iteration so the total $\sum \mu_j$ is constant. We found the iterative scheme converges in just 3 – 4 steps to a fixed set of mutation rates, unique up to a constant factor. The CI is computed by bootstrap parametrized by the uncertainties in data and the asymmetric constants.

The generation factor γ

This method does not give absolute mutation rates but *relative* mutation rates $\mu_j \gamma$, where γ is universal time scale constant. To find γ we apply our method to compute the $T = TMRCA$ of three famous DNA projects and choose γ so the scaled T/γ best fits the historical record. We choose the DNA projects for the O’Niall(M222), Gp A of Macdonald (R1a1a) and Gp A of the Hamiltons (I1). These are large groups with characteristic DNA and fairly accurate times of origin. Of course finding one constant γ from three projects is inherently more accurate than using one project to find 33 different mutation rates. Actually assuming a generation of 30years these three projects yield $\gamma = 1$ with about 5% error, i.e. there is no actual need for this correction. This is a constant error (like uncalibrated ^{14}C dating).

Thus γ is related to the length of a generation. Early researchers use 25yrs for $t > 500ybp$ and 27yrs for $t < 500ybp$. Balaresque and al (2010) used 30yrs based on Fenner(2005) who sees a 30yr generation, from extensive field work with modern hunter-gatherers. Our theory allows any nominal generation as it really doesn’t matter, being included in the γ factor which we compute in years not generations. However to give actual mutation rates we need an actual generation so we take 30 years. This appears in our worksheet computation. However this now makes our mutation rates very similar to actual laboratory meiosis rates, see Table 8 . Conversely then, if actual laboratory meiosis rates are indeed the correct long term phylogenetic rates have we have shown it follows *a priori* that the long term generation time is 30 years, i.e giving theoretical proof of Fenner

TABLE 8: Rate of Mutation $\times 10^{-3}$ /generation, at DYS: Meiosis vs Phylogenetic

#	DYS	Hamilton ⁴	< SD ³	Burgella ^{2,m}	< SD ³	Burgella ^{2,r}	< SD ³	Chandler ²	NIST ²	FTDNA ⁴
1	393.	0.72	0.1	1.03	0.32	2.6	0.25	0.76	0.08	1.43
2	390.	2.52	0.44	2.12	0.41	4.66	0.42	3.11	2.4	5.32
3	19/394 ¹	0.	0.	2.19	0.41	2.84	0.17	1.51	2.38	1.45
4	391.	4.98	0.99	2.71	0.46	2.02	0.16	2.65	2.88	4.15
5	385a	1.27	0.16	0.	0.	0.	0.	0.	2.1	5.68
6	385b ¹	0.	0.	0.	0.	0.	0.	0.	2.1	5.68
7	426.	0.06	0.02	0.22	0.5	0.46	0.16	0.09	0	0.26
8	388.	0.22	0.05	0.22	0.5	0.46	0.16	0.22	0	0.25
9	439.	3.76	0.43	5.48	0.8	2.89	0.17	4.77	0	4.95
10	389-I	1.94	0.2	2.53	0.48	2.2	0.16	1.86	1.88	2.23
11	392.	0.36	0.1	0.43	0.21	0.48	0.16	0.52	0.58	1.59
12	389b	2.96	0.33	3.17	0.53	2.54	0.16	2.42	2.96	2.72
13	458.	7.99	0.68	6.88	1.06	4.78	0.5	8.14	10.8	6.3
14	459a	0.4	0.07	0.	0.	0.	0.	0.	0	0
15	459b ¹	0.	0.	0.	0.	0.	0.	0.	0	0
16	455.	0.16	0.03	0.	0.	2.14	0.23	0.16	0	0.46
17	454.	0.11	0.02	0.	0.	2.18	0.23	0.16	0	0.47
18	447.	3.8	0.58	4.55	3.24	0.74	0.3	2.64	0	4.
19	437.	1.	0.18	1.08	0.39	0.	0.	0.99	1.5	2.15
20	448.	1.16	0.25	1.65	0.56	0.44	0.16	1.35	1.8	2.71
21	449.	11.71	1.63	18.87	8.13	9.64	1.79	8.38	0	7.84
22	460.	2.63	0.35	3.82	2.02	2.49	0.24	4.02	0	0
23	GATA H4	3.93	0.39	0.	0.	2.19	0.16	2.08	2.51	0
24	YCA IIa	0.32	0.07	0.	0.	0.	0.	0.	0	0
25	YCA IIb	1.4	0.25	0.	0.	0.	0.	0.	0	0
26	456.	8.1	1.85	4.5	0.87	3.27	0.29	7.35	0	0
27	607.	2.15	0.28	0.	0.	3.73	0.26	4.11	0	4.1
28	576.	10.66	1.19	16.16	6.05	4.18	0.41	10.22	0	10.2
29	570.	4.6	0.94	12.57	5.44	4.2	0.41	7.9	0	7.9
30	CDYa	14.71	1.39	0.	0.	0.	0.	0.	0	35.3
31	CDYb ¹	0.	0.	0.	0.	0.	0.	0.	0	35.3
32	442.	2.9	0.33	0.	0.	1.93	0.16	3.24	0	0
33	438.	0.43	0.06	0.43	0.26	0.75	0.29	0.55	0.7	0
Mean		3.34	15%	4.53	52%	2.66	15%	3.17	2.48	6.35

1. We have values but too inaccurate to use, $SD > 33\%$
2. meiosis: Burgella averages^m 80 sources then uses Regression^r, Chandler uses 20 sources, NIST produced their own data
3. one standard deviation ($\times 10^{-3}$ /generation), mean is average of relative SD, i.e. %
4. phylogenetic: Reduced Singularities(Hamilton) and not(FTDNA)

TABLE 9: Asymmetric Constants

#	DYS	A _j	SD	SD to 0.5
1.	393	0.675	0.087	2.
2.	390	0.463	0.093	0.4
3.	19/394	0.973	0.032	14.7
4.	391	0.029	0.008	62.8
5.	385a	0.699	0.096	2.1
6.	385b	0.82	0.085	3.8
7.	426	0.37	0.232	0.6
8.	388	0.91	0.072	5.7
9.	439	0.734	0.359	0.7
10.	389-I	0.779	0.105	2.7
11.	392	0.954	0.04	11.2
12.	389b	0.703	0.325	0.6
13.	458	0.512	0.137	0.1
14.	459a	0.139	0.125	2.9
15.	459b	0.003	0.001	353.
16.	455	0.277	0.168	1.3
17.	454	0.962	0.03	15.4
18.	447	0.154	0.025	13.6
19.	437	0.09	0.09	4.6
20.	448	0.216	0.172	1.6
21.	449	0.518	0.15	0.1
22.	460	0.107	0.05	7.9
23.	GATAH4	0.17	0.198	1.7
24.	YCAIIa	0.195	0.163	1.9
25.	YCAIIb	0.19	0.175	1.8
26.	456	0.671	0.416	0.4
27.	607	0.243	0.103	2.5
28.	576	0.387	0.157	0.7
29.	570	0.448	0.077	0.7
30.	CDYa	0.37	0.181	0.7
31.	CDYb	0.258	0.082	2.9
32.	442	0.603	0.17	0.6
33.	438	0.715	0.215	1.
Mean		0.26 ¹	0.134 ²	

1. Mean difference from 0.5.

2. Showing average is 2 SD from 0.5

TABLE 10: Comparison of BEAST, RHO, KAPZ with HAM, different data

SNP	Batini N^1	BEAST T^2	<SD	RHO ⁵ T^2	<SD	Hamilton N^1	HAM T^2	<SD ³	KAPZ ⁶	Anatolian	<SD
G2a-L31	18.	15300.	1075.	15740.	668.	1221.	7654 ⁴ .	1000.	7599.	12000.	2500.
I1-M253	46.	4190.	400.	3460.	145.	2898.	4109.	460.	5811.		
J2-M172	23.	21500.	1275.	21400.	908.	2211.	15000.	2000.	16968.	14600.	2500.
J2a-M410	15.	15200.	1025.	15700.	668.	1742.	13000.	2100.			
J2b-M102	8.	12200.	1025.	11240.	478.	437.	5286.	750.			
R1a-M198	25.	5960.	648.	6340.	270.	1270.	6016.	741.	6275.		
R1b-M269	144.	5550.	438.	4890.	208.	677.	6000.	700.	9150.	5700.	1000.
R1b-S116	100.	4210.	273.	3320.	140.	971.	4700.	376.	5485.		

Table comparing Batini et al(2015) MYS clock with STR clock using different data:

1. N size of sample
2. TMRCA (ybp)
3. Hamilton includes likely error in method by simulations of random trees, for comparison multiply by 0.5.
4. Batini computes for G2a across Europe and Middle east, Hamilton has Western European separate from Anatolian.
5. RHO, a KAPZ which accounts for population growth from a small initial population when genetic drift is minimal.
6. KAPZ assuming constant large population, however as the classic KAPZ assumed nearly equal mutation rates we compute TMRCA for each marker then average. Using the classic formula is equivalent to weighting by mutation rates.

Supplementary Information (online)

1. Distribution about mode for each DYS of R1b1a2
2. Randomly generated Branching Spectrum
3. Complete Mathematica worksheet for R1b1a2
4. Complete Mathematica worksheet for computing Asymmetric constants
5. Complete Mathematica worksheet for computing Mutation rates
6. Summary of Mathematica worksheets for computing TMRCA of the SNP:
G2a2 (Western Europe), R1b1a2, R1a1a, I1, L21, U106, J2, P312
7. DNA data for R1b1a2

SI 1: distribution about mode for DYS markers 1-33 for R1b1a2

Real data (blue) shows deviation from ideal distribution(red), especially for markers

#: 1, 9, 13, 18, 21, 23, 26, 27, 28, 29, 30, 31

i.e. 33% have significant deviation. (Ideal distribution, red, showing natural asymmetry).

2.

SI 2 : Random generation of branching spectrum

The purpose of this is to show how branching produces distribution of times (spectrum) which are similar to actual experimental data. In each example the TMRCA is 200 generations but with different values of the branching constant a . Regarding this as pseudo R1b1a2 distributions we assume same size sample

In[102]:= **NN = 667**

Out[102]= 667

We are only using the following 29 markers

```
In[90]:= B = {1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1}; BB[j_] := B[[j]];

a = RandomReal[{10, 100}]; PP = Table[{N[CDF[NormalDistribution[0, 1], -4 + 0.01*j]], -4 + 0.01*j}], {j, 0, 800}];
P = Interpolation[PP];  $\mu_0$  = {{0.0007222714614464581, 0.13788186468741065},
{0.00252191560852714, 0.1756056404979328}, {0.001303572925590671, 0.5164708472957918},
{0.004977150289157533, 0.19905067084267644}, {0.0012681155803029092, 0.1255101496903105},
{0.0031258069210570354, 0.3393998666171529}, {0.00006473656291412204, 0.24415449408166703},
{0.0002152896050608881, 0.21578440036607627}, {0.00376011308999344, 0.1131751046613454},
{0.001936945271013519, 0.1020415865125584}, {0.00036223149405663986, 0.27307390090816736},
{0.0029597631499781647, 0.11002851401721785}, {0.00799224117747206, 0.08488468832249377},
{0.0003981048976878356, 0.17668990646337812}, {0.0029797070704232153, 0.468587145151736},
{0.0001632628725620396, 0.2052681911425411}, {0.00011437505872541055, 0.21543041501213994},
{0.0038036479009438554, 0.15153182376717278}, {0.0009952936025767742, 0.17989256605708312},
{0.0011639531081141532, 0.21419593995600164}, {0.011706339153923283, 0.13913678085449122},
{0.002633694999084507, 0.13361927292526554}, {0.003931568575211744, 0.09969847327454723},
{0.0003205903099024094, 0.23316631317350148}, {0.001403927013011494, 0.17575983913452337},
{0.008103373405530107, 0.22821728732821633}, {0.0021453934823239366, 0.13045197490734378},
{0.010656815015340831, 0.11132961262899865}, {0.004602824417133062, 0.20485809551134662},
{0.014709975691241446, 0.09483238700016704}, {0.013402087511949615, 2.0047212464841033},
{0.002902798923344392, 0.11208753770180445}, {0.0004330809176749126, 0.1410759986588208}};
LM = Table[Log[ $\mu_0$ [[k, 1]]], Log[ $\mu_0$ [[k, 1]] (1 +  $\mu_0$ [[k, 2]])] - Log[ $\mu_0$ [[k, 1]]], {k, 1, 33}];
R1 = RandomReal[.001, .999, 33]; RR1 = P[R1];
 $\mu_0$  = Table[Exp[LM[[k, 1]] + RR1[[k]] * LM[[k, 2]]], {k, 1, 33}];
 $\alpha_0$  = {{0.6745914608277626, 0.08694232120498067},
{0.4633463923194976, 0.09257247592051233}, {0.9732429155540429, 0.03210806731916896},
{0.029005874713712673, 0.007504732619953859}, {0.6986605998989859, 0.09607929351304731},
{0.8204509474124598, 0.08501099550968096}, {0.36962272579754907, 0.23225932151607045},
{0.9095608279666191, 0.07168980200986297}, {0.7342893319987556, 0.3592143789100898},
{0.7793282986568104, 0.10519171804607536}, {0.9541405214613279, 0.04040159342004441},
{0.7027506677978216, 0.324864137462508}, {0.5123675151828064, 0.13680022562157804},
{0.13942843122652832, 0.1251291716934138}, {0.003012211149311151, 0.0014079884245269245},
{0.27724063205232263, 0.168111704826231}, {0.9624408662066506, 0.030002459548768477},
{0.1535778862226848, 0.025435319456792817}, {0.09003122818864748, 0.08968025427835938},
{0.2162236572064707, 0.17223041445171394}, {0.5181231787952857, 0.1498778201043168},
{0.10698102436459021, 0.04970501476507513}, {0.1695877971324342, 0.19796499082775937},
{0.19452455589105372, 0.1634469119340402}, {0.18997530592440287, 0.1753663072102212},
{0.670560655538126, 0.4161540139941387}, {0.2434904118943821, 0.10322120592000128},
{0.38731398742561307, 0.15700463039879492}, {0.4475198867563232, 0.07720166353650969},
{0.36971174696953457, 0.18055883975068912}, {0.25842754010671165, 0.08189448404551788},
{0.6028285845146566, 0.17033792642407947}, {0.7146170608926542, 0.2149602710368213}};
 $\beta_0$  = {{0.0730601063628016, 0.8210570611614773}, {0.8633993803230922, 0.3214354507199268},
{36.37327966429828, 44.847329380116044}, {0.029872348306083308, 0.00795979749562579},
{2.318517258827697, 1.058078676841258}, {4.569508641725883, 2.6369923277765306},
{0.5863516039108377, 0.5844832329598421}, {10.057155627551547, 8.764844700688261},
{2.763492100344712, 5.087866766990414}, {3.5316186620811685, 2.160171647259052}, {20.80574293178513,
19.210570818180535}, {2.364179130669454, 3.676715245943597}, {1.0507247386828336, 0.5753095121313209},
{0.1620184029844821, 0.16896026493785557}, {0.0030213119789396607, 0.0014165092216908354},
{0.3835863557736589, 0.3218170971527762}, {25.62468217456789, 21.267954536340646},
{0.18144361273515774, 0.0355028480718689}, {0.0989388108445023, 0.10830383969849475},
{0.27587418170317324, 0.28036612883691786}, {1.0752191348402145, 0.6454539967102586},
{0.11979703375112495, 0.062327374462433534}, {0.20422122476863463, 0.2870786856126378},
{0.24150277617245755, 0.2519254887371585}, {0.23453026471150157, 0.2672696465245145},
{2.0354601442067457, 3.834450695178595}, {0.3218603117828401, 0.18035992879043908},
{0.6321573848212977, 0.4182505585198409}, {0.810019901221205, 0.2529259317372638},
{0.5865756583466968, 0.4545067414320327}, {0.3484858919166162, 0.14891809274721798},
{1.5178045574553751, 1.0798299946484247}, {2.5040637086712945, 2.639381621418316}};
LB = {{0.729025827901993, 0.3960604222913862}, {-0.14687791357807545, 0.3722905738010174},
{3.5938344299209577, 1.2329745844759596}, {-3.510822032298842, 0.26646038718036874},
{0.8409278687844486, 0.45636006064334855}, {1.5194056809037075, 0.577084438291061},
{-0.5338356626633742, 0.996813565549179}, {2.308284383892832, 0.871503338048881},
{0.10164951337181467, 1.8411005286954736}, {1.2617563103447185, 0.6116661661273677},
{3.035229051087664, 0.9233302017219662}, {0.8604308710679793, 1.5551762547292591},
{0.04948015338909639, 0.547535896844417}, {-1.8200453515319506, 1.0428461324485363},
{-5.802064111820329, 0.468839110811709}, {-0.9581905054957993, 0.8389690934227843},
{3.243556034437963, 0.829979251701657}, {-1.7068103471817775, 0.19566876748475165}};
```

```

{-2.3132536922107123`, 1.0946547545301615`}, {-1.2878103805965928`, 1.0162825934127382`},
{0.07252448716876952`, 0.6002999535589346`}, {-2.1219563536146264`, 0.5202747723446721`},
{-1.5885514375725887`, 1.4057240423363127`}, {-1.4208743104174564`, 1.0431577339602012`},
{-1.450170638819988`, 1.139595552212786`}, {0.7107219084237044`, 1.8838249946048182`},
{-1.1336376419013978`, 0.5603670977368852`}, {-0.4586168892212833`, 0.6616240964076923`},
{-0.21069646220858051`, 0.3122465649991398`}, {-0.5334536194697443`, 0.7748476005859001`},
{-1.054157531906251`, 0.42732889968139737`}, {0.41726492065504456`, 0.7114420557932557`},
{0.9179148956742922`, 1.0540393250692588`}}; nn = 60; τ = .01;
R2 = RandomReal[ {.001, .999}, 33]; RR2 = P[R2];

β = Table[ Exp[LB[[j, 1]] + LB[[j, 2]]*RR2[[j]]], {j, 1, 33}]; α = Table[  $\frac{\beta[[j]]}{1 + \beta[[j]]}$ , {j, 1, 33}];

I0[t_] := N[ 1 +  $\sum_{k=1}^{nn} \left( \frac{(t/2)^{(2*k)}}{k! * (k!)}$  ), nn]; I1[t_] := N[  $\frac{t}{2} + \sum_{k=1}^{nn} \left( \frac{(t/2)^{(2*k+1)}}{k! * ((k+1)!)} \right)$ , nn];
I2[t_] := N[  $\frac{t^2}{8} + \sum_{k=1}^{nn} \left( \frac{(t/2)^{(2*k+2)}}{k! * ((k+2)!)} \right)$ , nn]; EE[t_] := N[ 1 +  $\sum_{k=1}^{nn} \left( \frac{t^k}{k!} \right)$ , nn];
μ = Table[ {μ00[[j]]*α[[j]], μ00[[j]]*(1-α[[j]])}, {j, 1, 33}];

H[t_, j_] := EE[-t*μ00[[j]]] *  $\left( I0[2*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}] + \right.$ 

$$\left( \sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}} * I1[2*\tau*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}] * \left( \frac{\mu[[j, 2]]}{\mu[[j, 1]]} * I2[2*(1-\tau)*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}] \right) \right.$$


$$\left. + \left( \sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}} * I1[2*\tau*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}] * \left( \frac{\mu[[j, 1]]}{\mu[[j, 2]]} * I2[2*(1-\tau)*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}] \right) \right) \right];

G0[x_] := UnitStep[200 - x]*UnitStep[x]*Exp[(x - 200)/a];
pp = NIntegrate[ G0[x], {x, 1, 200}]; G[x_] := G0[x]/pp; PP1 = Table[ {NIntegrate[ G[t], {t, 0, j}], j}, {j, 0, 200}];
P1 = Interpolation[PP1]; R3 = RandomReal[ {.001, .999}, 33]; RR3 = P1[R3];

RR4 = Table[ 1 - H[ RR3[[k]], k], {k, 1, 33}]; RR5 = Table[  $\sqrt{\frac{RR4[[k]] (1 - RR4[[k]])}{NN}}$ , {k, 1, 33}];

TS[k_] := Normal[InverseSeries[Series[ (1 - H[x, k]), {x, 0, 60}]]]; TSS[t_, k_] := TS[k] /. x -> t;
ZZZ001 = Parallelize[ Table[ TSS[ RR4[[k]] + RR5[[k]], k], {k, 1, 33} ] ];
ZZZ002 = Parallelize[ Table[ TSS[ RR4[[k]] - RR5[[k]], k], {k, 1, 33} ] ]; RR6 = 0.5*(ZZZ001 - ZZZ002);

g[j_] := RR3[[j]]; s[j_] := RR6[[j]]; f[x_, j_] :=  $\frac{1}{\sqrt{2*\pi*s[j]^2}} * Exp[-\frac{(x - g[j])^2}{2*s[j]^2}]$ ;

F[x_] := Sum[ BB[j]*f[x, j], {j, 1, 33}]/29;
Plot[ {F[x], G[x]}, {x, 0, 450}, Exclusions -> None, PlotRange -> {0, .03}, PlotLabel -> {a}]$$

```

Each of the following shows the Branching Spectrum (Blue) resulting a spreadout generation of branches (density function in Red)

For a fast expansion the mean is almost exactly the TMRCA

The above distribution is similar to actual R1a2a

This example shows Spectrum similar to J2 or G2a2.

SI 3 : Mathematica worksheet for R1b1a2

Use FTDNA R1b1a2 data, which has been corrected(for missing files etc) and taken from R1b1a2data file

δδ;

Each file has NN members

NN = Length[δδ]

677

We use asymptotic rates α_0 , (and their logs etc β_0 , LB shown)

0.674591	0.0869423	2.07306	0.821057	0.729026	0.39606
0.463346	0.0925725	0.863399	0.321435	-0.146878	0.372291
0.973243	0.0321081	36.3733	44.8473	3.59383	1.23297
0.0290059	0.00750473	0.0298723	0.0079598	-3.51082	0.26646
0.698661	0.0960793	2.31852	1.05808	0.840928	0.45636
0.820451	0.085011	4.56951	2.63699	1.51941	0.577084
0.369623	0.232259	0.586352	0.584483	-0.533836	0.996814
0.909561	0.0716898	10.0572	8.76484	2.30828	0.871503
0.734289	0.359214	2.76349	5.08787	1.0165	1.8411
0.779328	0.105192	3.53162	2.16017	1.26176	0.611666
0.954141	0.0404016	20.8057	19.2106	3.03523	0.92333
0.702751	0.324864	2.36418	3.67672	0.860431	1.55518
0.512368	0.1368	1.05072	0.57531	0.0494802	0.547536
0.139428	0.125129	0.162018	0.16896	-1.82005	1.04285
0.00301221	0.00140799	0.00302131	0.00141651	-5.80206	0.468839
0.277241	0.168111	0.383586	0.321817	-0.958191	0.838969
0.962441	0.0300025	25.6247	21.268	3.24356	0.829979
0.153578	0.0254353	0.181444	0.0355028	-1.70681	0.195669
0.0900312	0.0896803	0.0989388	0.108304	-2.31325	1.09465
0.216224	0.17223	0.275874	0.280366	-1.28781	1.01628
0.518123	0.149878	1.07522	0.645454	0.0725245	0.6003
0.106981	0.049705	0.119797	0.0623274	-2.12196	0.520275
0.169588	0.197965	0.204221	0.287079	-1.58855	1.40572
0.194525	0.163447	0.241503	0.251925	-1.42087	1.04316
0.189975	0.175366	0.23453	0.26727	-1.45017	1.1396
0.670561	0.416154	2.03546	3.83445	0.710722	1.88382
0.24349	0.103221	0.32186	0.18036	-1.13364	0.560367
0.387314	0.157005	0.632157	0.418251	-0.458617	0.661624
0.44752	0.0772017	0.81002	0.252926	-0.210696	0.312247
0.369712	0.180559	0.586576	0.454507	-0.533454	0.774848
0.258428	0.0818945	0.348486	0.148918	-1.05416	0.427329
0.602829	0.170338	1.5178	1.07983	0.417265	0.711442
0.714617	0.21496	2.50406	2.63938	0.917915	1.05404

```
Table[1, {j, 1, 33}]; BB[j_] := B[[j]]; BB0 = Table[1, {j, 1, 33}];
PP = Table[{N[CDF[NormalDistribution[0, 1], -4 + 0.01*j]], -4 + 0.01*j}, {j, 0, 800}];
P = Interpolation[PP];
```

We bootstrap with n02 cycles

n02 = 300

300

The method of reduction is applied


```

Z = Flatten[Table[ {ClearAll[L1, δ, m0, f1, m1, m2, m3, m4, m5, m6, m, f2, f3, mm,
    δ1, m10, m11, m12, m13, m14, m16, m17, n, R, RR, β, α]; n = IntegerPart[NN*.5];
    L1 = RandomSample[Range[NN], n]; δ = Table[δδ[L1[[j]]], {j, 1, n}]; m0 = Mean[δ];
    m1 = Table[Commonest[Table[δ[k, j]], {k, 1, n}][[1]], {j, 1, 33}];
    m2 = Table[m1[[j]], {i, n}, {j, 33}]; m3 = δ - m2; m4 = Abs[m3]; f2[u_] := UnitStep[u - 1];
    m5 = f2[m4]; f3[j_] := N[Sum[m5[[i, j]], {i, 1, n}]/n]; m6 = Table[f3[j], {j, 1, 33}];
    m = Table[Sum[m5[[i, j]]*BBO[j], {j, 33}], {i, n}]; mm = N[Sum[m[[i]], {i, n}]/n];
    f1[k_] := 1 - UnitStep[Abs[k] - 0.5]; δ1 = Transpose[δ];
    f2[j_, k_] := N[Sum[f1[δ1[[j, i]] - k], {i, 1, n}]/n]; R = RandomReal[{.001, .999}, 33]; RR = P[R];

    β = Table[Exp[LB[[j, 1]] + LB[[j, 2]]*RR[[j]]], {j, 1, 33}]; α = Table[ $\frac{\beta[[j]]}{1 + \beta[[j]]}$ , {j, 1, 33}];

    m10 = Parallelize[
        Table[{Max[.001, f2[j, m1[[j]] - 2]], Max[.001, f2[j, m1[[j]] - 1]], Max[.001, f2[j, m1[[j]]]},
            Max[.001, f2[j, m1[[j]] + 1]], Max[.001, f2[j, m1[[j]] + 2]]}, {j, 1, 33}]; m11 = Parallelize[Table[
            {Min[.999, Max[.001, β[[j]]*m10[[j, 2]]], Min[.999, Max[.001, m10[[j, 4]]/β[[j]]]}], {j, 1, 33}];
        m12 = Parallelize[Table[{m10[[j, 4]] - m11[[j, 1]], m10[[j, 2]] - m11[[j, 2]]}, {j, 1, 33}];
        m13 = Parallelize[Table[{UnitStep[m12[[j, 1]] - .001], UnitStep[m12[[j, 2]] - .001]}, {j, 1, 33}];
        m14 = Parallelize[Table[m13[[j, 1]] - m13[[j, 2]], {j, 1, 33}];
        m15 = Parallelize[
            Table[{m10[[j, 2]] - m13[[j, 2]]*m11[[j, 2]], m10[[j, 3]] - m13[[j, 1]]*m10[[j, 5]]/β[[j]] -
                m13[[j, 2]]*m10[[j, 1]]*β[[j]], m10[[j, 4]] - m13[[j, 1]]*m11[[j, 1]]}, {j, 1, 33}];
            m16 = Parallelize[Table[{Min[.999, Max[.001, m15[[j, 1]]], Min[1, Max[.001, m15[[j, 2]]]},
                Min[.999, Max[.001, m15[[j, 3]]]}], {j, 1, 33}]; m17 =
            Parallelize[Table[{m16[[j, 1]] + m16[[j, 3]]/(m16[[j, 2]] + m16[[j, 1]] + m16[[j, 3]]}, {j, 1, 33}];
            {Table[{k, m17[[k]], 1 - m10[[k, 3]], m14[[k]]}, {k, 1, 33}], {q2, 1, n02}], 2];

    MM[j_] := Mean[1.0*Table[Z[[k, j, 2]], {k, 1, n02}]];
    MM0[j_] := Mean[1.0*Table[Z[[k, j, 3]], {k, 1, n02}]];
    MM1[j_] := Mean[1.0*Table[Z[[k, j, 4]], {k, 1, n02}]];
    SS[j_] := (Variance[1.0*Table[Z[[k, j, 2]], {k, 1, n02}]])^(.5);
    SS0[j_] := (Variance[1.0*Table[Z[[k, j, 3]], {k, 1, n02}]])^(.5);
    SS1[j_] := (Variance[1.0*Table[Z[[k, j, 4]], {k, 1, n02}]])^(.5);
    ZZ = Table[{j, MM[j], MM0[j], MM1[j], SS[j], SS0[j], SS1[j]}, {j, 1, 33}]; MatrixForm[ZZ]

```

1	0.271316	0.302637	1.	0.0178978	0.0126611	0.
2	0.201682	0.300628	0.476	0.0271493	0.0130464	0.869741
3	0.178373	0.116323	-0.96	0.195153	0.00940206	0.280562
4	0.385975	0.352135	1.	0.029537	0.0130741	0.
5	0.157559	0.223067	0.992	0.0278212	0.0111525	0.126491
6	0.263609	0.394063	-0.024	0.0879698	0.0135655	0.989617
7	0.141216	0.177767	0.4	0.0227142	0.010168	0.909565
8	0.0349038	0.0394798	-0.952	0.00624372	0.00567578	0.279299
9	0.394868	0.403013	-0.152	0.1575	0.0127973	0.990363
10	0.184642	0.219704	-0.876	0.0224796	0.0107642	0.479106
11	0.226154	0.251632	-0.628	0.0615687	0.0110867	0.77201
12	0.351466	0.378108	-0.424	0.115257	0.0130506	0.907479
13	0.509485	0.587614	0.936	0.0981115	0.014322	0.352706
14	0.0633869	0.078296	0.648	0.0132519	0.0072736	0.747214
15	0.442594	0.251704	1.	0.164795	0.0110868	0.
16	0.0197234	0.0253991	0.256	0.00523943	0.00399652	0.926227
17	0.0098425	0.0118655	-0.844	0.00292294	0.00312497	0.395334
18	0.291304	0.317184	1.	0.0184938	0.013626	0.
19	0.127226	0.135211	0.932	0.0204479	0.00974217	0.357614
20	0.0994612	0.1147	0.968	0.0157589	0.00802896	0.251454
21	0.587612	0.66217	0.112	0.111483	0.012874	0.995702
22	0.322642	0.352448	0.992	0.0211385	0.0139905	0.126491
23	0.564598	0.446161	0.944	0.232074	0.0135255	0.330607
24	0.0302449	0.0624574	0.104	0.00865472	0.00665072	0.934171
25	0.0591171	0.122888	-0.18	0.0195145	0.00921815	0.96713
26	0.522952	0.525399	0.852	0.220455	0.0129397	0.52075
27	0.730244	0.518251	1.	0.169299	0.0147761	0.
28	0.653017	0.661103	0.188	0.163942	0.0107722	0.982097
29	0.365704	0.465498	0.984	0.0420302	0.0138983	0.178526
30	0.744854	0.682332	0.652	0.170513	0.0129874	0.757092
31	0.920049	0.693498	0.936	0.130351	0.0114332	0.352706
32	0.170861	0.234063	-0.376	0.0308018	0.0119116	0.919787
33	0.0521186	0.0713901	-0.228	0.0115408	0.00636655	0.961099

This output is for each marker (j), reduced frequency f0, unreduced frequency f0, mean singular branch ± and SD for each.

Next we conservatively weight the reduction by mean ±

```
ZZZ = Table[ Min[ ZZ[[k, 3]], Abs[ZZ[[k, 4]] * ZZ[[k, 2]] + (1 - Abs[ZZ[[k, 4]]) * ZZ[[k, 3]] ], {k, 1, 33}];
```

This has SD

```
ZZS = Table[
  ((Abs[ZZ[[k, 4]]] * (ZZ[[k, 5]]^2 + ZZ[[k, 2]]^2) + (1 - Abs[ZZ[[k, 4]]]) * (ZZ[[k, 6]]^2 + ZZ[[k, 3]]^2) -
  (Abs[ZZ[[k, 4]]] * ZZ[[k, 2]] + (1 - Abs[ZZ[[k, 4]]]) * ZZ[[k, 3]] )^2)^(.5) , {k, 1, 33}];
```

So we display reduced frequency with SD

```
MatrixForm[Table[ {k, ZZZ[[k]], ZZS[[k]] }, {k, 1, 33}]]
```

1	0.271316	0.0178978
2	0.25353	0.053684
3	0.116323	0.191605
4	0.352135	0.029537
5	0.158083	0.0283351
6	0.390932	0.02764
7	0.163147	0.0242701
8	0.0351235	0.00629413
9	0.401775	0.0625938
10	0.18899	0.0243017
11	0.235632	0.0507733
12	0.366811	0.0768374
13	0.514486	0.0968948
14	0.0686349	0.0135322
15	0.251704	0.164795
16	0.0239461	0.00500464
17	0.0101581	0.00304516
18	0.291304	0.0184938
19	0.127769	0.0200045
20	0.0999489	0.0158004
21	0.65382	0.0457388
22	0.322881	0.0212574
23	0.446161	0.227143
24	0.0591073	0.0120047
25	0.111409	0.0271748
26	0.523314	0.203551
27	0.518251	0.169299
28	0.659583	0.071813
29	0.367301	0.0435678
30	0.682332	0.141076
31	0.693498	0.137793
32	0.210299	0.0371815
33	0.0669962	0.0112707

To compute times for each marker need mutation rates(and SD)

$$\mu_0 = \begin{pmatrix} 0.0007222714614464581 & 0.13788186468741065 \\ 0.00252191560852714 & 0.1756056404979328 \\ 0.001303572925590671 & 0.5164708472957918 \\ 0.004977150289157533 & 0.19905067084267644 \\ 0.0012681155803029092 & 0.1255101496903105 \\ 0.0031258069210570354 & 0.3393998666171529 \\ 0.00006473656291412204 & 0.24415449408166703 \\ 0.0002152896050608881 & 0.21578440036607627 \\ 0.00376011308999344 & 0.1131751046613454 \\ 0.001936945271013519 & 0.1020415865125584 \\ 0.00036223149405663986 & 0.27307390090816736 \\ 0.0029597631499781647 & 0.11002851401721785 \\ 0.00799224117747206 & 0.08488468832249377 \\ 0.0003981048976878356 & 0.17668990646337812 \\ 0.0029797070704232153 & 0.468587145151736 \\ 0.0001632628725620396 & 0.2052681911425411 \\ 0.00011437505872541055 & 0.21543041501213994 \\ 0.0038036479009438554 & 0.15153182376717278 \\ 0.0009952936025767742 & 0.17989256605708312 \\ 0.0011639531081141532 & 0.21419593995600164 \\ 0.011706339153923283 & 0.13913678085449122 \\ 0.002633694999084507 & 0.13361927292526554 \\ 0.003931568575211744 & 0.09969847327454723 \\ 0.0003205903099024094 & 0.23316631317350148 \\ 0.001403927013011494 & 0.17575983913452337 \\ 0.008103373405530107 & 0.22821728732821633 \\ 0.0021453934823239366 & 0.13045197490734378 \\ 0.010656815015340831 & 0.11132961262899865 \\ 0.004602824417133062 & 0.20485809551134662 \\ 0.014709975691241446 & 0.09483238700016704 \\ 0.013402087511949615 & 2.0047212464841033 \\ 0.002902798923344392 & 0.11208753770180445 \\ 0.0004330809176749126 & 0.1410759986588208 \end{pmatrix};$$

Also need hyperbolic Bessel functions

$$\begin{aligned} nn &= 60; \tau = .01; \\ I0[t_] &:= N\left[1 + \sum_{k=1}^{nn} \left(\frac{(t/2)^{(2*k)}}{k! * (k!)}\right), nn\right]; I1[t_] := N\left[\frac{t}{2} + \sum_{k=1}^{nn} \left(\frac{(t/2)^{(2*k+1)}}{k! * ((k+1)!)}\right), nn\right]; \\ I2[t_] &:= N\left[\frac{t^2}{8} + \sum_{k=1}^{nn} \left(\frac{(t/2)^{(2*k+2)}}{k! * ((k+2)!)}\right), nn\right]; EE[t_] := N\left[1 + \sum_{k=1}^{nn} \left(\frac{t^k}{k!}\right), nn\right]; \\ SGN[x_] &:= UnitStep[x] + (-1) * (1 - UnitStep[x]); \mu00 = Table[\mu0[[j, 1]], {j, 1, 33}]; \\ \alpha &= Table[\alpha0[[j, 1]], {j, 1, 33}]; \\ \mu &= Table[\{\mu00[[j]] * \alpha[[j]], \mu00[[j]] * (1 - \alpha[[j]])\}, {j, 1, 33}]; H[t_, j_] := \\ &EE[-t * \mu00[[j]]] * \left(I0\left[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}\right] + \left(\sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}} * I1\left[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}\right] * \right. \right. \\ &\left. \left(\frac{\mu[[j, 2]]}{\mu[[j, 1]]} \right) * I2\left[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}\right] \right. \\ &\left. + \left(\sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}} * I1\left[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}\right] * \left(\frac{\mu[[j, 1]]}{\mu[[j, 2]]} \right) * \right. \right. \\ &\left. \left. I2\left[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}\right] \right) \right]; \end{aligned}$$

The expansion times τ_k for marker k are obtained by use of the inverse function

$$TS[k_] := Normal[InverseSeries[Series[(1 - H[x, k]), {x, 0, 60}]]]; TSS[t_, k_] := TS[k] /. x -> t;$$

which is applied to give expansion times for each marker and SD

$$\begin{aligned} ZZZ01 &= Parallelize[Table[TSS[ZZZ[[k]], k], {k, 1, 33}]]; \\ ZZZ001 &= Parallelize[Table[TSS[(ZZZ[[k]] + ZS[[k]]), k], {k, 1, 33}]]; \\ ZZZ002 &= Parallelize[Table[TSS[(ZZZ[[k]] - ZS[[k]]), k], {k, 1, 33}]]; ZZZ02 = 0.5 * (ZZZ001 - ZZZ002); \end{aligned}$$

So we display expansion times (generations) with SD for each marker

MatrixForm[**Table**[{**k**, **zzz01**[**k**]], **zzz02**[**k**]], {**k**, 1, 33}]]

```
( 1 473.619 39.963
 2 125.825 33.9499
 3 95.1727 170.344
 4 88.307 9.3999
 5 140.995 28.7181
 6 172.164 17.2196
 7 2875.86 490.739
 8 166.571 30.4803
 9 153.758 35.9503
10 112.347 16.7275
11 750.745 188.169
12 172.623 52.4247
13 116.11 45.7151
14 180.157 37.1396
15 97.3955 75.2826
16 149.185 31.7161
17 89.301 26.9175
18 94.9748 7.57
19 138.922 23.583
20 92.1457 15.6538
21 142.072 32.3578
22 154.002 12.9221
23 165.236 141.401
24 191.894 40.5842
25 85.7219 22.6325
26 113.641 105.139
27 402.666 259.62
28 153.753 54.4328
29 114.045 20.0969
30 122.051 115.054
31 123.396 84.7769
32 86.5257 18.4559
33 162.454 28.7201)
```

We are only using the following 29 markers

```
UU[x_] := 1 - UnitStep[-x];
B = UU[{1, 2, 0, 4, 5, 0, 7, 8, 9, 10, 11, 12, 13, 14, 0, 16, 17, 18, 19, 20, 21, 22, 23,
 24, 25, 26, 27, 28, 29, 30, 0, 32, 33}]; BB[j_] := B[j];
```

We first sketch the spectrum of times

```
QQ[x_, q3_, q4_] := N[  $\frac{1}{q4 \sqrt{2} \pi} \text{Exp}\left[-\frac{(x - q3)^2}{2 * (q4)^2}\right]$  ];
Q[x_, j_] := BB[j] * QQ[x, zzz01[j] + .01, zzz02[j] + .01];
Pic1[a_, b_] := Plot[Evaluate[Table[Q[x, j], {j, 1, 33}]], {x, 0, a}, PlotRange → {0, b}, PlotStyle → Table[
  RGBColor[UnitStep[22 - j] * (22 - j) / 22, Sin[j *  $\pi$  / 33], UnitStep[j - 11] * (j - 11) / 22 ], {j, 1, 33}],
  PlotLabel →
  Times]
```

This plots the time distributions for each DYS marker, showing error spread and outliers, over time to 1000 generations

Pic1[1000, .06]

The individual branches may be combined to obtain average times

Q0[x_] := Sum[Q[x, j] / 29, {j, 1, 33}]; Pic2[a_, b_] := Plot[Q0[x] , {x, 1, a}, PlotRange -> {0, b}]

Pic2[1000, .01]

We compute the mean, in generations

```
m1 = Sum[BB[j] * ZZZ01[[j]], {j, 1, 33}] / Sum[BB[j], {j, 1, 33}]
```

269.5

or in years

```
30 * m1
```

8085.01

One observes outliers are exaggerating the mean.

For later use we generate random quintiles at roughly 0.30,0.40,0.50,0.60,0.70%

```
n01 = 10 000 000; w0 = Flatten[Parallelize[Table[{Clear[R, Z]; R = RandomReal[ {.0005, .9995}, 29]; Z = Sort[R];
  {Z[[9]], Z[[12]], Z[[15]], Z[[18]], Z[[21]]}, {k, 1, n01}], 1]; {Mean[w0], Sqrt[Variance[w0]]}
{{0.300213, 0.400082, 0.499979, 0.599873, 0.699791}, {0.0822449, 0.0879039, 0.0896917, 0.0878771, 0.0822073}}
```

We compute the same quintiles and SD for our data via bootstrap

```
{n01, n00} = {100 000, 33}; PPP[x_] := N[InverseCDF[NormalDistribution[0, 1.0], x]]; DD = Clear[ LM, WWO, Z];
LM = Table[ Log[ 1 + (.001 + ZZZ02[[j]]) / (.1 + ZZZ01[[j]]) ], {j, 1, 33}]; WWO = Flatten[Parallelize[
  Table[{Clear[RR, R0, R1, RR, Z]; R0 = RandomReal[ {.01, .99}, n00]; R1 = Table[ PPP[R0[[j]]], {j, 1, n00}];
    RR = Table[ BB[j]*ZZZ01[[q1, j]]*Exp[ LM[[j]]*R1[[j]]], {j, 1, n00}]; Z = Sort[RR];
    {Z[[13]], Z[[16]], Z[[19]], Z[[22]], Z[[25]]}], {k, 1, n01}], 1];
D0 = {NN, Mean[WWO], Sqrt[Variance[WWO] ]}
```

```
{677, {109.93175892632237, 126.23981628986326,
  142.17092881048893, 157.77587953279243, 177.7456766471929},
  {8.375335520073904, 9.32935978135218, 9.501609160384556, 10.31777176434264, 13.910536924726323}}
```

The stochastic simulation uses the branching distribution τ_0 to generate random quintiles. To speed the process we compile a distribution function F4.

```
Clear[ PP, nn, n00]; {nn, n00,  $\tau$ } = {60, 33, .01};
{11, 12} = {IntegerPart[ D0[[2, 1]] / 3], 2*IntegerPart[D0[[2, 5]]]}; l = IntegerPart[12 / 11] + 1;
PP[x_] := N[InverseCDF[NormalDistribution[0, 1.0], x]]; LM = Table[ Log[ 1 +  $\mu$ 0[[j, 2]] ], {j, 1, 33}];
I0[t_] := BesselI[0, t]; I1[t_] := BesselI[1, t]; I2[t_] := BesselI[2, t];
```

$$\begin{aligned} \text{HH}[t_, j_] := & 1 - \text{EE}[-t * \mu 0[[j]]] * \left(I_0[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}] + \left(\sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}} \right) * \right. \\ & I_1[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}] * \left(\frac{\mu[[j, 2]]}{\mu[[j, 1]]} \right) * I_2[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}] \\ & + \left(\sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}} \right) * I_1[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}] * \left(\frac{\mu[[j, 1]]}{\mu[[j, 2]]} \right) * \\ & \left. I_2[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]}] \right); \end{aligned}$$

$$\text{G0}[j_] := \text{D}[\text{HH}[x, j], x]; \text{G}[t_, j_] := \text{G0}[j] /. x \rightarrow t; \text{S0}[t_, j_] := \frac{\sqrt{\text{HH}[t, j] (1 - \text{HH}[t, j])}}{\text{G}[t, j]};$$

```
S1 = Parallelize[Table[{j, k^2, S0[k^2, j]}, {j, 1, n00}, {k, 0, 40}];
S2[j_] := Interpolation[Join[Table[{S1[[j, k, 2]], S1[[j, k, 3]]}, {k, 1, 41}],
  {{101600, S1[[j, 41, 3]] + 100000*(S1[[j, 41, 3]] - S1[[j, 40, 3]]/79)}, InterpolationOrder -> 1];
S[j_, x_] := If[x > 0, S2[j][x], 0]; n000 = Sum[BB[j], {j, 1, n00}];
v[t1_, m0_] := N[Sum[ (t1^2*BB[j]* $\mu$ 0[[j, 2]]^2 + m0*BB[j]*S[j, t1]^2)/n000, {j, 1, n00}];
f[t_, x_, T_,  $\alpha$ _, m0_] := If[x < T && x > 0, \frac{\text{Exp}[-(T-x)/\alpha - (x-t)^2/(2 v[x, m0])]}{\alpha * \sqrt{2 * \pi * v[x, m0]}}, 0];
```

$$\text{F}[t_, T_, \alpha_, m0_] := \text{If} \left[\text{NIntegrate} \left[\frac{\text{Exp}[-(T-x)/\alpha - (x-t)^2/(2 v[x, m0])]}{\alpha * \sqrt{2 * \pi * v[x, m0]}} , \{x, 0.001, T\} \right] > .0000001, \right.$$

$$\left. \text{NIntegrate} \left[\frac{\text{Exp}[-(T-x)/\alpha - (x-t)^2/(2 v[x, m0])]}{\alpha * \sqrt{2 * \pi * v[x, m0]}} , \{x, 0.001, T\} \right], .0000001 \right];$$

```
 $\sigma$  = N[Sum[ (BB[j]* $\mu$ 0[[j, 2]]^2)/n000, {j, 1, n00}];
ff[t1_] := N[Sum[ BB[j]*S[j, t1]^2/n000, {j, 1, n00}];
data = Table[{11 + i*11, ff[11 + i*11]}, {i, 0, 1}]; Clear[a, b, c, d];
FF = FindFit[data, a*x + b*x^2 + c*x^3 + d*x^4, {a, b, c, d}, x]; {a, b, c, d} = {a, b, c, d} /. FF;
GG[t_] :=  $\sigma$ *t + b*t^2 + c*t^3 + d*t^4; m = 1.0/NN; Clear[P,  $\alpha$ ];
vv[t_, m0_] :=  $\sigma$ *t^2 + m0*(GG[t]); P[x_] := N[CDF[NormalDistribution[0, 1.0], x]];
Q2[u_, x_, m0_] := N[P[(u-x)/\sqrt{vv[x, m0]}] - P[-x/\sqrt{vv[x, m0]}]];
```

$$\begin{aligned} \text{H2}[u_, T_, \alpha_, m0_] := & \frac{\text{Min}[1, 4 * \alpha / T]}{1000} * \frac{T}{\alpha} * \\ & \text{Sum} \left[\text{Exp} \left[-\frac{\text{Min}[1, 4 * \alpha / T]}{1000} * j * \frac{T}{\alpha} \right] * Q2 \left[u, \left(1 - \frac{\text{Min}[1, 4 * \alpha / T]}{1000} * j \right) * T, m0 \right], \{j, 0, 999\} \right]; \text{Clear}[X]; \\ X = & \text{Parallelize}[\text{Table}[\{.001 + .1 * k * 1.2^{(k/10)}, 11 + i * 11, .001 + j * .1, \end{aligned}$$

```
H2[ (.001 + .1*k*1.2^(k/10)) * (11 + i*11), 11 + i*11, (11 + i*11) * (.001 + j*.1), m]],
{k, 0, 23}, {i, 0, 1}, {j, 0, 14}]; X1 = Flatten[X, 2]; n1 = Length[X1];
X2 = Table[{X1[[j, 1]], X1[[j, 2]], X1[[j, 3]], X1[[j, 4]], {j, 1, n1}};
Clear[F10]; F10 = Interpolation[X2, InterpolationOrder -> 1];
F1[x_, y_, a_] :=
Which[ .001 < x && x < 4, F10[x, y, a], x < .001, F10[.001, y, a], x >= 4, F10[4, y, a]]; X0 =
Parallelize[Table[{X[[k, i, j, 1]], X[[k, i, j, 2]], X[[k, i, j, 3]], X[[k, i, j, 4]] + 1 - X[[24, i, j, 4]]},
{k, 1, 24}, {i, 1, 1 + 1}, {j, 1, 15}]]; f4[{x1_, x2_, x3_, x4_}] := x4;
X00 = Table[ Select[Table[X0[[k, i, j]], {k, 1, 24}], f4[#] > .025*n - .001 &][[1]],
{i, 1, 1 + 1}, {j, 1, 15}, {n, 1, 40}];
X01 = Table[ DeleteDuplicates[X00[[i, j]], {i, 1, 1 + 1}, {j, 1, 15}];
X05 = Table[ Table[{X01[[i, j, k, 4]], X01[[i, j, k, 1]]}, {k, 1, Length[X01[[i, j]]]}],
{i, 1, 1 + 1}, {j, 1, 15}]; X07 = Parallelize[Table[
{Clear[W1, F04, F05]; WX = Table[{X01[[i, j, k, 4]], X01[[i, j, k, 1]]}, {k, 1, Length[X01[[i, j]]]}];
F04 = Interpolation[WX, InterpolationOrder -> 1]; F05[x_] := Which[ 0 <= x && x < X01[[i, j, 1, 4]],
F04[X01[[i, j, 1, 4]]], x < 1 && x >= X01[[i, j, 1, 4]], F04[x]];
{Table[{0.0499*n, F05[0.0499*n], X01[[i, j, 1, 2]], X01[[i, j, 1, 3]]}, {n, 0, 20}]],
{i, 1, 1 + 1}, {j, 1, 15}]; X08 = Flatten[X07, 4]; n2 = Length[X08];
X09 = Table[{X08[[j, 1]], X08[[j, 3]], X08[[j, 4]], X08[[j, 2]]}, {j, 1, n2}];
Clear[F20];
F20 = Interpolation[X09, InterpolationOrder -> 1];
Clear[F4]; F4[u1_, u2_, u3_] :=
Which[u1 < 0.01, F20[.01, u2, u3], u1 > .99, F20[.99, u2, u3], True, F20[u1, u2, u3]];
Our data quintiles are b1, b2, b3, b4, b5. We search for T between L1 and L2
```

```
{m1, b1, b2, b3, b4, b5, L1, L2} =
{1.0/NN, D0[[2, 1]], D0[[2, 2]], D0[[2, 3]], D0[[2, 4]], D0[[2, 5]], D0[[2, 2]], 2*D0[[2, 5]] }
{0.0014771, 109.932, 126.24, 142.171, 157.776, 177.746, 126.24, 355.491}
```

We generate n01 random (T,β) and random quintiles W0 (on (0,1)) which are transformed to distribution quintiles by F4. These are filtered so they are within r SD of the data quintiles. In this case r=2

```
n01 = 5 000 000; Clear[W10]; r = 2;
W10 = Flatten[Parallelize[Table[{Clear[Y, a, J, Z, Z0, Z1]; Y = RandomReal[{L1, L2}];
a = RandomReal[{.05, 1.3}]; J = RandomInteger[{1, 10 000 000}]; Z0 = W0[[J]];
Z1 = {Y*F4[Z0[[1]], Y, a], Z0[[2]], Z0[[3]], Z0[[4]], Z0[[5]]}; Z = Join[Z1, {Y, a}];
{Z}}, {k, 1, n01}]], 2]; W11 = Select[W10, #[[1]] < D0[[2, 1]] + r*D0[[3, 2]] &&
#[[1]] > D0[[2, 1]] - r*D0[[3, 2]] &]; n02 = Length[W11];
W12 = Flatten[Parallelize[Table[{Clear[Y, a, Z, Z0]; Z0 = W11[[k]]; Y = Z0[[6]]; a = Z0[[7]];
Z = {Z0[[1]], Y*F4[Z0[[2]], Y, a], Z0[[3]], Z0[[4]], Z0[[5]], Z0[[6]], Z0[[7]]};
{Z}}, {k, 1, n02}]], 2]; W13 = Select[W12, #[[2]] < D0[[2, 2]] + r*D0[[3, 2]] &&
#[[2]] > D0[[2, 2]] - r*D0[[3, 2]] &]; n03 = Length[W13];
W14 = Flatten[Parallelize[Table[{Clear[Y, a, Z, Z0]; Z0 = W13[[k]]; Y = Z0[[6]]; a = Z0[[7]];
Z = {Z0[[1]], Z0[[2]], Y*F4[Z0[[3]], Y, a], Z0[[4]], Z0[[5]], Z0[[6]], Z0[[7]]};
{Z}}, {k, 1, n03}]], 2]; W15 = Select[W14, #[[3]] < D0[[2, 3]] + r*D0[[3, 3]] &&
#[[3]] > D0[[2, 3]] - r*D0[[3, 3]] &]; n04 = Length[W15];
W16 = Flatten[Parallelize[Table[{Clear[Y, a, Z, Z0]; Z0 = W15[[k]]; Y = Z0[[6]]; a = Z0[[7]];
Z = {Z0[[1]], Z0[[2]], Z0[[3]], Y*F4[Z0[[4]], Y, a], Y*F4[Z0[[5]], Y, a], Z0[[6]], Z0[[7]]};
{Z}}, {k, 1, n04}]], 2]; Clear[W20]; W20 = Select[W16,
#[[4]] < D0[[2, 4]] + r*D0[[3, 4]] && #[[4]] > D0[[2, 4]] - r*D0[[3, 4]] &&
#[[5]] < D0[[2, 5]] + r*D0[[3, 5]] && #[[5]] > D0[[2, 5]] - r*D0[[3, 5]] &]; n05 = Length[W20]
```

207 321

which is the number filtered from 5 million random distributions to be within 2SD of the data quintiles. This sample of random data near the data quintiles has mean and SD:

```
{Mean[Table[W20[[j, 6]], {j, 1, n05}]], Sqrt[Variance[Table[W20[[j, 6]], {j, 1, n05}]]}]
{194.816, 32.3391}
```

We use a quasilinear estimator on these


```

WW1 = Table[ { W20[[j, 1]], W20[[j, 2]], W20[[j, 3]], W20[[j, 4]], W20[[j, 5]], 1} * (W20[[j, 6]]) ^ (-1),
  {j, 1, n05}]; WW2 = Table[ 1, {j, 1, n05}];
WW3 = LeastSquares[WW1, WW2]; BB[{B1_, B2_, B3_, B4_, B5_}] :=
  WW3[[1]] * B1 + WW3[[2]] * B2 + WW3[[3]] * B3 + WW3[[4]] * B4 + WW3[[5]] * B5
+ WW3[[6]]; BBW = Table[
  BB[{W20[[j, 1]], W20[[j, 2]], W20[[j, 3]], W20[[j, 4]], W20[[j, 5]]}] / W20[[j, 6]] - 1, {j, 1, n05}];
D1 = {n01, n05, Mean[BBW], Sqrt[Variance[BBW]]}; D2 = {1 + D1[[3]], 1 / (1 + D1[[3]])};
BB1[{B1_, B2_, B3_, B4_, B5_}] = D2[[2]] * BB[{B1, B2, B3, B4, B5}];

BBW = Table[ BB1[{W20[[j, 1]], W20[[j, 2]], W20[[j, 3]], W20[[j, 4]], W20[[j, 5]]}] / W20[[j, 6]] - 1,
  {j, 1, n05}]; D3 = {n01, n05, Mean[BBW], Sqrt[Variance[BBW]]}

{5 000 000, 207 321, -2.09898 × 10-17, 0.107177}

```

Thus we estimate the TMRCA from our experimental data in generations

```

t1 = BB1[D0[[2]]]

200.15147075754703`

```

Also we estimate the SD given the variance in the experimental data

```

ee = {{1, 0, 0, 0, 0}, {0, 1, 0, 0, 0}, {0, 0, 1, 0, 0}, {0, 0, 0, 1, 0}, {0, 0, 0, 0, 1}}; WS =
  Flatten[Table[ {Clear[b1, b2, b3, b4, b5]; {b1, b2, b3, b4, b5} = D0[[2]] + (-1)^j * (D0[[3]].ee[[i]]) ee[[i]];
    {BB1[{b1, b2, b3, b4, b5}]}, {i, 1, 5}, {j, 0, 1}]; D4 = {Mean[WS], Sqrt[Variance[WS]]}

{200.151, 9.33904}

```

Thus our estimate for the TMRCA in ybp is

```

30 * t1

6004.54

```

One should not forget that the overall variance is the sum of the variance from sample error and the intrinsic error of the stochastic simulation

```

r0 = Sqrt[D3[[4]]^2 + (D4[[2]] / D4[[1]])^2]

0.116893

r0 * 30 * t1

701.89

```

ie haplotype R1b1a2 we get TMRCA= 4004BC± 701(one SD). The SD will be improved by compared with independent data from Poland, Germany, Scandinavia.

SI 4: Mathematica worksheet for computing asymmetric constants

Asymmetric Rates

We use iteration to compute the asymmetric rates, using the SNP groups G2a3, R1b1a2, R1a1, I1, L21,U106, J2, P312.

As usual we first download the data:

Next we compute asymmetric ratios for each sample, ie comparing the modal, compute mean & SD by bootstrap

```
n0 = 60; τ = .01; B = Table[1, {j, 1, 33}]; BB[j_] := B[[j]];
B0 = Table[0, {j, 1, 33}]; BB0[j_] := 1 - B0[[j]]; nn = 500;
ZZ = Flatten[Table[{ClearAll[δ0, n01, AAA, AA, VV]; δ0 = δδ[[q1]]; n01 = Count[Flatten[δ0], _?Positive]/33;
  AAA = Flatten[Table[{ClearAll[L1, δ, m0, f1, m1, m2, m3, m4, f2, m5, f3, m6, m, mm,
    δ1, m10, n, β]; n = IntegerPart[n01*.5]; L1 = RandomSample[Range[n01], n];
    δ = Table[δ0[[L1[[j]]]], {j, 1, n}]; m0 = Mean[δ];
    m1 = Table[Commonest[Table[δ[[k, j]], {k, 1, n}]][[1]], {j, 1, 33}];
    m2 = Table[m1[[j]], {i, n}, {j, 33}]; m3 = δ - m2; m4 = Abs[m3]; f2[u_] := UnitStep[u - 1];
    m5 = f2[m4]; f3[j_] := N[Sum[m5[[i, j]], {i, 1, n}]/n]; m6 = Table[f3[j], {j, 1, 33}];
    m = Table[Sum[m5[[i, j]]*BB[j]*BB0[j], {j, 33}], {i, n}]; mm = N[Sum[m[[i]], {i, n}]/n];
    f1[k_] := 1 - UnitStep[Abs[k] - 0.5]; δ1 = Transpose[δ];
    f2[j_, k_] := N[Sum[f1[δ1[[j, i]] - k], {i, 1, n}]/n]; m10 =
      Parallelize[Table[{j, Max[.001, f2[j, m1[[j]] - 1]], Max[.001, f2[j, m1[[j]] + 1]]}, {j, 1, 33}]];
    β = Table[
      {Table[{k, β[[k]]}, {k, 1, 33}], {q2, 1, nn}}, {j, 1, 33}];
    AA[k_] := Mean[1.0*Table[AAA[[q2, 1, k, 2]], {q2, 1, nn}]];
    VV[k_] := (1.0*Variance[Table[AAA[[q2, 1, k, 2]], {q2, 1, nn}]]).^5;
    {Table[{k, AA[k], VV[k]}, {k, 1, 33}], {q1, 1, 8}}, 2];
  {Table[{k, β[[k]]}, {k, 1, 33}], {q2, 1, nn}}, {j, 1, 33}];
  AA[k_] := Mean[1.0*Table[AAA[[q2, 1, k, 2]], {q2, 1, nn}]];
  VV[k_] := (1.0*Variance[Table[AAA[[q2, 1, k, 2]], {q2, 1, nn}]]).^5;
  {Table[{k, AA[k], VV[k]}, {k, 1, 33}], {q1, 1, 8}}, 2];
}, {q1, 1, 8}], 2];
ZZ = {{1, 0.34124004932308, 0.04070614049345421, {2, 1.3425832045163308, 0.2072917242173085},
{3, 2.8669743455505783, 0.5579673443314054}, {4, 6.238244300144298, 2.6992475924137485},
{5, 0.17675358385412288, 0.030837714919849317}, {6, 11.449443875924946, 2.055617661830038},
{7, 0.5563833060109307, 0.4707562782744052}, {8, 0.015387188065282453, 0.0060982527061936515},
{9, 8.465801503896417, 1.4771148554720825}, {10, 10.234275324675327, 4.114535248528078},
{11, 1.1121506300888655, 0.3961348566038753}, {12, 0.8925961847501103, 0.09123730461722532},
{13, 2.912520259326451, 0.3424303279379609}, {14, 0.014366076320080993, 0.001577177917044884},
{15, 30.967213114754063, 8.721983260309754}, {16, 0.7452017177267171, 0.34817859260664175},
{17, 25.032431693989068, 13.477421678655078}, {18, 0.9935365859841375, 0.11059630477645156},
{19, 0.19678737045062492, 0.04898738451633803}, {20, 0.7010893189277894, 0.07186215400538538},
{21, 1.005924611669342, 0.10694281590792797}, {22, 12.738072257427222, 2.525018347467305},
{23, 0.15271083946129096, 0.020890134592999387}, {24, 0.005957260660382156, 0.001524381051080261},
{25, 35.381577595628414, 22.929770682903666}, {26, 2.37652549555331, 0.30639164053097684},
{27, 2.205921965891891, 0.2803511100783317}, {28, 1.0128193767169247, 0.09833844886254253},
{29, 0.5222437709203244, 0.04513181922720796}, {30, 0.8052978036777253, 0.0769079292196483},
{31, 1.1710763888817453, 0.41445045967815847}, {32, 6.244882795165085, 1.407819226565891},
{33, 11.445996877439505, 7.466288839874486}, {{1, 69.68722463768114, 85.09915119865201},
{2, 0.9507346989631812, 0.16000503009556366}, {3, 3.7784792207792197, 1.7032588400213144},
{4, 0.07337807490680874, 0.0227271247958635}, {5, 8.547523809523812, 4.315376197112916},
{6, 4.043625577608854, 0.9564379551189195}, {7, 0.8545789646339647, 0.2109247364824565},
{8, 2.470104140786751, 3.367788478385527}, {9, 1.2192302241047765, 0.18725651850325808},
{10, 1.3380109756661225, 0.3067651337682906}, {11, 10.95134996549344, 10.264049007301262},
{12, 0.9138022272784421, 0.1525136471644648}, {13, 3.273218039023715, 1.140103506160712},
{14, 0.6044294427794437, 0.26416013872610394}, {15, 0.028836471642804437, 0.015481420285993891},
{16, 0.2730679299516913, 0.2850621746777503}, {17, 6.443966086956507, 5.293125003516419},
{18, 0.6602044030191525, 0.12242662725614636}, {19, 0.8072098231628835, 0.2296070116747267},
{20, 1.7204248986307809, 0.4663017827097156}, {21, 1.2495566610107984, 0.2045056287054631},
{22, 0.46069352851827544, 0.08600094759661286}, {23, 2.228482605267968, 0.3927721031972498},
{24, 0.39337459956709997, 0.2276232731604596}, {25, 0.18304185532441464, 0.09028530101393667},
{26, 14.355423783182584, 7.3228500270752}, {27, 4.379797799685772, 0.8136273361636982},
{28, 1.0923754901478637, 1.573814626515883}, {29, 1.9985779859856976, 0.33860995597805205},
{30, 1.0559897881027172, 0.3647672130715871}, {31, 1.057677753583667, 0.6765748751118308},
{32, 0.7495914191634647, 0.14067546435346648}, {33, 1.612758874458874, 0.6666644473215361}},
{{1, 3.246506379731381, 0.9906245538868559}, {2, 0.4620230597438728, 0.055689765051482654},
{3, 0.7665905191007474, 6.07481272176514}, {4, 26.11678867793558, 64.83844601742037},
{5, 0.8531357444083333, 0.12101159803841736}, {6, 2.4889035228190557, 0.3167496456895208},
{7, 0.908623376623376, 0.45087892258743856}, {8, 7.0967139107611565, 3.9225686083115594},
{9, 124.11480577427824, 59.959197376819695}, {10, 6.803626465168365, 1.3004929336844337},
{11, 10.106569553805775, 5.459937308655013}, {12, 0.7020089103061474, 0.06553582820393634},
{13, 2.96759790512103, 0.3287844512138404}, {14, 0.32329066692586816, 0.10491326383889364},
{15, 0.12642685041088653, 0.026458878126077556}, {16, 1.2136236652236654, 0.4808947382367965},
```

```
{17, 8.801569553805779, 4.625331889700318}, {18, 0.7060918152792786, 4.461488402147883},
{19, 5.901842519685039, 2.318850370125652}, {20, 0.6283198606783605, 0.09692420897638346},
{21, 0.734559732974123, 0.0639716624823721}, {22, 1.1769364716576647, 0.13064244559249322},
{23, 8.310380320565166, 1.2564165380703907}, {24, 0.8717271243253679, 0.7033806368386137},
{25, 0.4700131294288555, 0.08654245197320244}, {26, 0.48431947419153343, 0.04567033170565495},
{27, 0.338089217285014, 0.036228796596728556}, {28, 0.6749341806870721, 0.0561050499821062},
{29, 0.8518837564255255, 0.07201178023111929}, {30, 2.0472169578223953, 0.18955193862107528},
{31, 1.851323560771722, 0.27034626895354436}, {32, 11.832568987000819, 2.9337125092668863},
{33, 2.259105738705739, 0.6602739715045449}, {{1, 2.790717092831961, 0.37669454459134116},
{2, 32.922264691210565, 7.474162362183765}, {3, 16.2980904796317, 3.315199770903363},
{4, 10.195169416625685, 2.4666737391825233}, {5, 9.30401975465309, 0.9785696718029584},
{6, 3.5045263704259546, 0.2934999796891092}, {7, 0.4825151996894418, 0.2598622170956618},
{8, 4.791941205624368, 1.1621531944487533}, {9, 3.506003915115374, 0.3494920307368679},
{10, 22.42677337348642, 9.016930212954351}, {11, 9.720252657004822, 4.291760976525525},
{12, 5.7306379690342295, 0.7593948888156394}, {13, 1.850056408539215, 0.11887149234671737},
{14, 0.9657402894582913, 0.14672916907121953}, {15, 1.4883593677407867, 0.32565076179071214},
{16, 9.715665976535544, 1.7817210954955038}, {17, 4.687830986887512, 2.132499136276013},
{18, 0.4358652799165425, 0.030083500086338505}, {19, 0.2802648404354452, 0.05503952589221627},
{20, 0.8932598861320162, 0.09967916134174723}, {21, 4.3475176724364255, 0.339877792547306},
{22, 6.774072550631299, 0.7444354494930424}, {23, 0.6406951613437628, 0.06350959800789847},
{24, 0.9055449355552639, 0.2302868825338679}, {25, 0.7282300176053865, 0.14591387184078866},
{26, 4.048898986996067, 0.3800683489072291}, {27, 2.058878333595069, 0.21146438788440788},
{28, 3.282699746595748, 0.23824129609838465}, {29, 1.9369813769312854, 0.42201905215252433},
{30, 1.0333486475839748, 0.06061011744000764}, {31, 0.8411910360596747, 0.046323590411132015},
{32, 1.0281221402075844, 0.08527689372651796}, {33, 1.2034508730381543, 0.26718177472306875},
{{1, 2.0722741754633405, 0.5965899379840204}, {2, 1.4168144106742486, 0.17531630028548906},
{3, 22.364534370946803, 16.297016454803117}, {4, 0.07598363762426734, 0.014026198750760423},
{5, 2.62813504583054, 0.5882276305968261}, {6, 4.3323458266190045, 0.6813889563259711},
{7, 1.803939892532889, 1.7725749635937749}, {8, 14.44357976653687, 3.6058379553867823},
{9, 0.6605010840871939, 0.0680097794029452}, {10, 4.159758549528874, 0.8614974565060681},
{11, 8.422036112655174, 4.843487079743514}, {12, 7.752496068896335, 1.644622632061394},
{13, 1.258608653588706, 0.12326188267869516}, {14, 0.9934277999777993, 0.3505526569435658},
{15, 0.022644809992945616, 0.010486698487494617}, {16, 1.2663213897213905, 0.5551395319115999},
{17, 6.681073929961092, 3.17685090749435}, {18, 0.7468183879667457, 0.0835639970340969},
{19, 0.4482547641302482, 0.09165882944713141}, {20, 0.9533503140976208, 0.1590815005169476},
{21, 1.642682907577223, 1.4092132091191896}, {22, 0.24375875446888265, 0.039065730702633444},
{23, 0.8398703336480671, 0.09843141669134386}, {24, 1.1559387102711383, 0.7800068868300222},
{25, 2.44558437847176, 0.3606221835473245}, {26, 18.40106631349627, 14.42255629137895},
{27, 0.7047251800511043, 0.0914513908078258}, {28, 1.1392628934999125, 0.10758734404821567},
{29, 1.2059422965996023, 0.12994991678190518}, {30, 1.9842927810520068, 0.283086744368526},
{31, 0.8222599045722995, 0.08219256104179412}, {32, 1.2024593950935827, 0.16991464710370885},
{33, 1.6772957431457418, 0.6854911556765044}, {{1, 1.46715095096327775, 0.322735621677224},
{2, 9.794276827972602, 1.4903424701472743}, {3, 12.146019841269846, 5.911815884151131},
{4, 0.1651276481643472, 0.019981266912756143}, {5, 2.134565342158425, 0.3754769383461285},
{6, 2.200829621032871, 0.29296438400570096}, {7, 1.5559066310249068, 0.9063164587985859},
{8, 9.820253263707569, 3.795247030345671}, {9, 0.4661922880405847, 0.044512065100786855},
{10, 2.4497793648575383, 0.3866324230966719}, {11, 3.879183793983793, 1.309948998819321},
{12, 1.1012509687094012, 0.11623342849303527}, {13, 1.2453776151988567, 0.1002195073898197},
{14, 0.9131839077589069, 0.34723355453260785}, {15, 0.025838000392480253, 0.010254716138099801},
{16, 1.028157714507715, 0.49096314659944446}, {17, 4.962748041775456, 2.894326312710387},
{18, 0.3451456877067722, 0.029928109548563155}, {19, 0.34104845200016837, 0.0549746472694613},
{20, 0.9575322206080765, 0.19556541492962293}, {21, 1.9419776427015725, 0.15260438491415762},
{22, 0.22319015860737448, 0.027992228425512973}, {23, 0.1591388352704033, 0.019620325198378256},
{24, 0.5943649424305143, 0.21998651278571796}, {25, 0.5802564272055627, 0.08836721460476803},
{26, 0.8130169744953265, 0.05841639617671167}, {27, 0.5039067956348808, 0.06638468768312722},
{28, 2.5174698985501345, 0.2367365360972958}, {29, 2.198452594801344, 0.198092274793078},
{30, 0.678116437169044, 0.5958228580694431}, {31, 1.040325010030788, 0.08533948267172964},
{32, 1.3733009207248386, 0.13160206143403966}, {33, 0.31107557956608417, 0.06829841187591698}},
{{1, 33.58391397849462, 22.692234826829658}, {2, 1.9159309731196754, 0.18800142031658548},
{3, 26.083029553779554, 7.744793992843321}, {4, 0.6714018082860438, 0.08628075530739998},
{5, 2.415550853104411, 0.24701599938513483}, {6, 1.4053755217319501, 0.5218577386009037},
{7, 2.123477526881721, 1.640478772901343}, {8, 3.149311402388486, 0.3831282051669624},
{9, 10.131995869320795, 1.8398531873461956}, {10, 0.8583772513959186, 0.07425292778866129},
{11, 7.156490476190473, 4.196594109935358}, {12, 8.419027063129796, 1.4231458099832917},
{13, 0.9129300939028936, 0.646453193400261}, {14, 2.765120236856604, 28.186691785452968},
{15, 1.279706130866659, 0.28994412686385396}, {16, 0.4196545956612151, 0.15220727416723492},
{17, 10.81464523809524, 4.643136160527687}, {18, 0.42547055064762207, 0.0511369577579925},
{19, 0.7340483664674416, 0.06492890129506346}, {20, 1.2581869753995263, 0.2511429972281426},
{21, 1.954455564904273, 0.5374323647312577}, {22, 4.575758194476958, 0.5376818815028099},
{23, 4.581032301169236, 0.5326312552701141}, {24, 1.427663200688202, 0.406678466606991},
{25, 0.731572652764317, 0.10419370960204741}, {26, 2.447620099379591, 0.27859682688782134},
{27, 0.6830185844925737, 0.06269429155083044}, {28, 1.209659900230228, 0.8680263522563889},
{29, 0.9682059561064948, 0.08529465551084889}, {30, 1.4936271137498116, 0.48752588066681773},
{31, 0.78382576788020266, 0.08193484174363924}, {32, 4.609653202851833, 0.6114529237350141},
{33, 8.109344572452637, 5.082284849199681}}, {{1, 1.6735628292799711, 0.3472843687851052},
{2, 0.8278117159034468, 0.10259960765129912}, {3, 9.257416017316011, 2.8973192234659315},
{4, 0.10312713057135606, 0.019739575990286785}, {5, 2.7708839244319803, 0.5069556299067204},
{6, 2.53372499773155, 0.35511292228057895}, {7, 3.3236887628866025, 2.1771740049673665},
```

```
{8, 22.705154639175195, 4.618690988275637}, {9, 0.6932923404030238, 0.07142619386258775},
{10, 1.6422443459489213, 0.22343189677769743}, {11, 6.933031320569463, 4.482358547636128},
{12, 4.627444877818076, 0.9939280675024266}, {13, 1.146616427897921, 0.12070110287573059},
{14, 1.1234586030928133, 0.3574183284274497}, {15, 0.06495773854095792, 0.01932828786115451},
{16, 3.6866756013745703, 2.732388089113388}, {17, 6.3114209621993105, 4.431256587710841},
{18, 0.7879332258740601, 0.09498990371033132}, {19, 0.11353336545434571, 0.021878483471024297},
{20, 0.26024822930858593, 0.03831764143129113}, {21, 2.16175433867254, 0.2437471966276382},
{22, 0.18705003337128243, 0.030174515892025914}, {23, 0.21720709412486067, 0.03192612435677606},
{24, 3.914706578301425, 2.5755067548202737}, {25, 0.3103829538825348, 0.0857221247243353},
{26, 0.24479287004816036, 0.029592232869885386}, {27, 0.6428400300377158, 0.09449249208255014},
{28, 0.83554182468699171, 0.08238329408971676}, {29, 1.3243225547048012, 0.13093536663589436},
{30, 1.1438769050416202, 0.11195004285009096}, {31, 0.7082906766097438, 0.07179481424095362},
{32, 0.9967939265787041, 0.14024637318903613}, {33, 1.8243247863247858, 0.756888723762818}};
```

We take log means so we can compare different SNP

```
ZZ1 = Table[ Sum[ Log[ZZ[[q1, j, 2]]], {q1, 1, 8}]/8, {j, 1, 33}]
```

```
{1.31373, 0.757191, 1.99988, -0.280674, 0.763736, 1.19072, 0.177269, 1.18514,
1.14768, 1.28868, 1.80974, 0.874402, 0.556616, -0.609292, -1.46434, 0.196938, 2.06596,
-0.506093, -0.71244, -0.199719, 0.498861, 0.0973345, -0.177881, -0.618401, -0.0527832,
0.799548, 0.00593251, 0.245115, 0.220906, 0.17746, -0.0109586, 0.758582, 0.74493}
```

```
ZZ2 = Table[ Log[ZZ[[q1, j, 2]] + ZZ[[q1, j, 3]]] - Log[ZZ[[q1, j, 2]]], {q1, 1, 8}, {j, 1, 33};
```

$$Q[x_, q3_, q4_] := N \left[\frac{1}{q4 \sqrt{2 \pi}} \text{Exp} \left[-\frac{(x - q3)^2}{2 * (q4)^2} \right] \right];$$

```
ZZ0 = Table[ Log[ZZ[[q1, j, 2]]] - ZZ1[[q1]], {q1, 1, 8}, {j, 1, 33};
```

```
Q0[x_, q1_, j_] := Q[x, ZZ0[[q1, j]], ZZ2[[q1, j]]];
```

```
ZZ4 = Table[ Min[ Table[ZZ0[[q1, j]], {q1, 1, 8}], {j, 1, 33}];
```

```
ZZ5 = Table[ Max[ Table[ZZ0[[q1, j]] + 3.0*ZZ2[[q1, j]], {q1, 1, 8}], {j, 1, 33}];
```

```
ZZ6 = Table[ Max[ Table[Q0[ZZ0[[q1, j]], q1, j], {q1, 1, 8}], {j, 1, 33}];
```

We use colors for our SNP j=1,2,3,4,5,6,7,8

```
{RGBColor[1, 0, 0]}, {RGBColor[0, 0, 1]}, {RGBColor[0, 1, 0]}, {RGBColor[0, 0, 0]}, {RGBColor[1, 0, 1]}, {RGBColor[0, 1, 1]}, {RGBColor[1, 1, 0]},
{RGBColor[.5, 0, 1]}
```

```
In[8]:= {Style[1 - G2, RGBColor[1, 0, 0]], Style[2 - R1b1, RGBColor[0, 0, 1]],
Style[3 - R1a, RGBColor[0, 1, 0]], Style[4 - I1, RGBColor[0, 0, 0]],
Style[5 - L21, RGBColor[1, 0, 1]], Style[6 - U106, RGBColor[0, 1, 1]],
Style[7 - J2, RGBColor[1, 1, 0]], Style[8 - P312, RGBColor[.5, 0, 1]] }
```

```
Out[8]:= {1 - G2, 2 - R1b1, 3 - R1a, 4 - I1, 5 - L21, 6 - U106, 7 - J2, 8 - P312}
```

Pictures show the distribution of mean asymmetric ratios for a given marker, across the 8 different SNP. If there is no asymmetry mutation and no singular branches these should be randomly distributed about 0.

```
mm1[{j_, x1_, x2_, x3_, x4_, x5_, x6_, x7_, x8_}] :=
(x1*ZZ0[[1, j]] + x2*ZZ0[[2, j]] + x3*ZZ0[[3, j]] + x4*ZZ0[[4, j]]
+ x5*ZZ0[[5, j]] + x6*ZZ0[[6, j]]
+ x7*ZZ0[[7, j]] + x8*ZZ0[[8, j]])/(x1 + x2 + x3 + x4
+ x5 + x6 + x7 + x8); vv1[{j_, x1_, x2_, x3_, x4_, x5_, x6_, x7_, x8_}] :=
(x1*(ZZ2[[1, j]]^2 + ZZ0[[1, j]]^2) + x2*(ZZ2[[2, j]]^2 + ZZ0[[2, j]]^2) +
x3*(ZZ2[[3, j]]^2 + ZZ0[[3, j]]^2) + x4*(ZZ2[[4, j]]^2 + ZZ0[[4, j]]^2)
+ x5*(ZZ2[[5, j]]^2 + ZZ0[[5, j]]^2) + x6*(ZZ2[[6, j]]^2 + ZZ0[[6, j]]^2)
+ x7*(ZZ2[[7, j]]^2 + ZZ0[[7, j]]^2) + x8*(ZZ2[[8, j]]^2 + ZZ0[[8, j]]^2))/(x1 + x2 + x3 + x4
+ x5 + x6 + x7 + x8) - mm1[{j, x1, x2, x3, x4, x5, x6, x7, x8}]^2;
ms[{j_, x1_, x2_, x3_, x4_, x5_, x6_, x7_, x8_}] :=
{mm1[{j, x1, x2, x3, x4, x5, x6, x7, x8}], vv1[{j, x1, x2, x3, x4, x5, x6, x7, x8}]^(.5)}
```

```
Pic[{j_, x1_, x2_, x3_, x4_, x5_, x6_, x7_, x8_}] :=
Plot[{Q0[x, 1, j]*x1, Q0[x, 2, j]*x2, Q0[x, 3, j]*x3, Q0[x, 4, j]*x4, Q0[x, 5, j]*x5, Q0[x, 6, j]*x6,
Q0[x, 7, j]*x7, Q0[x, 8, j]*x8}, {x, ZZ4[[j]], ZZ5[[j]]}, PlotRange -> {0, ZZ6[[j]]},
PlotStyle -> {{RGBColor[1, 0, 0]}, {RGBColor[0, 0, 1]}, {RGBColor[0, 1, 0]}, {RGBColor[0, 0, 0]},
{RGBColor[1, 0, 1]}, {RGBColor[0, 1, 1]}, {RGBColor[1, 1, 0]}, {RGBColor[.5, 0, 1]}}, PlotLabel ->
{j, mm1[{j, x1, x2, x3, x4, x5, x6, x7, x8}], vv1[{j, x1, x2, x3, x4, x5, x6, x7, x8}]^(.5)}];
```

The curve is for DYS k=1 (first entry) and SNP j=1,2..8 (next eight)

Pic[{1, 1, 1, 1, 1, 1, 1, 1, 1}]

Discarding the outlier SNP, choose only SNP j = 3,5,6,8

x = {1, 0, 0, 1, 0, 1, 1, 0, 1}

Pic[**x**]

This shows how many (4) of marker 1 are considered, what is their mean and SD

ms1 = {1, Sum[x[[j]], {j, 2, 9}], ms[x]}

{1, 4, {-0.584705, 0.39606}}

Pic[{2, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {2, 1, 1, 0, 0, 1, 0, 0, 1}

Pic[x]

ms2 = {2, Sum[x[[j]], {j, 2, 9}], ms[x]}

{2, 4, {-0.904069, 0.372291}}

Pic[{3, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {3, 1, 1, 0, 1, 1, 1, 1, 1}

Pic[x]

ms3 = {3, Sum[x[[j]], {j, 2, 9}], ms[x]}

{3, 7, {1.59396, 1.23297}}

Pic[{4, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {4, 0, 1, 0, 0, 1, 2, 0, 1}

Pic[**x**]

ms4 = { 4, Sum[x[[j]], {j, 2, 9}], ms[x]}

{4, 5, {-3.23015, 0.26646}}

Pic[{5, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {5, 0, 0, 0, 0, 1, 1, 1, 1}

Pic[x]

ms5 = {5, Sum[x[[j]], {j, 2, 9}], ms[x]}

{5, 4, {0.0771924, 0.45636}}

Pic[{6, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {6, 1, 1, 0, 0, 1, 1, 1, 1}

Pic[**x**]

ms6 = { 6, Sum[x[[j]], {j, 2, 9}], ms[x]}

{6, 6, {0.328683, 0.577084}}

Pic[{7, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {7, 1, 1, 1, 1, 1, 1, 1, 1}

Pic[**x**]

ms7 = {7, Sum[x[[j]], {j, 2, 9}], ms[x]}

{7, 8, {-0.711104, 0.996814}}

Pic[{8, 1, 1, 1, 1, 1, 1, 1, 1}]

X = {8, 0, 1, 1, 1, 1, 1, 1, 1}

Pic[X]

ms8 = {8, Sum[X[[j]], {j, 2, 9}], ms[X]}

{8, 7, {1.12315, 0.871503}}

Pic[{9, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {9, 1, 1, 1, 1, 1, 1, 1, 1}

Pic[x]

ms9 = {9, Sum[x[[j]], {j, 2, 9}], ms[x]}

{9, 8, {-0.131182, 1.8411}}

Pic[{10, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {10, 1, 1, 1, 0, 1, 1, 1, 1}

Pic[X]

ms10 = { 10, Sum[X[[j]], {j, 2, 9}], ms[X]}

{10, 7, {-0.0269243, 0.611666}}

Pic[{11, 1, 1, 1, 1, 1, 1, 1, 1}]

X = {11, 0, 1, 1, 1, 1, 1, 1, 1}

Pic[X]

```
ms11 = {11, Sum[X[[j]], {j, 2, 9}], ms[X]}
```

```
{11, 7, {1.22549, 0.92333}}
```

```
Pic[{12, 1, 1, 1, 1, 1, 1, 1, 1}]
```

```
x = {12, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
Pic[X]
```

```
ms12 = {12, Sum[X[[j]], {j, 2, 9}], ms[X]}
```

```
{12, 8, {-0.0139712, 1.55518}}
```


Pic[{13, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {13, 1, 1, 1, 1, 0, 1, 1, 1, 1}

Pic2[**x**]

ms13 = { 13, Sum[x[[j]], {j, 2, 9}], ms[x]}

{13, 7, {-0.507136, 0.547536}}

Pic[{14, 1, 1, 1, 1, 1, 1, 1, 1}]

X = {14, 0, 1, 1, 1, 1, 1, 0, 1}

Pic[X]

ms14 = { 14, Sum[X[[j]], {j, 2, 9}], ms[X]}

{14, 6, {-1.21075, 1.04285}}

Pic[{15, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {15, 0, 1, 1, 0, 1, 1, 0, 1}

Pic[x]

ms15 = {15, Sum[x[[j]], {j, 2, 9}], ms[x]}

{15, 5, {-4.33772, 0.468839}}

Pic[{16, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {16, 1, 1, 1, 0, 1, 1, 1, 1}

Pic[X]

ms16 = { 16, Sum[X[[j]], {j, 2, 9}], ms[X]}

{16, 7, {-1.15513, 0.838969}}

Pic[{17, 1, 1, 1, 1, 1, 1, 1, 1}]

X = {17, 1, 1, 1, 1, 1, 1, 1, 1}

Pic[X]

ms17 = { 17, Sum[X[[j]], {j, 2, 9}], ms[X]}

{17, 8, {1.17759, 0.829979}}

Pic[{18, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {18, 1, 1, 0, 0, 1, 0, 1, 1}

Pic[X]

```
ms18 = {18, Sum[X[[j]], {j, 2, 9}], ms[X]}
```

```
{18, 5, {-1.20072, 0.195669}}
```

```
Pic[{19, 1, 1, 1, 1, 1, 1, 1, 1}]
```

```
X = {19, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
Pic[X]
```

```
ms19 = {19, Sum[X[[j]], {j, 2, 9}], ms[X]}
```

```
{19, 8, {-1.60081, 1.09465}}
```

Pic[{20, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {20, 1, 1, 1, 1, 1, 1, 1, 1}

Pic[**x**]

ms20 = { 20, Sum[**x**[**j**]], {j, 2, 9}}, **ms**[**x**]

{20, 8, {-1.08809, 1.01628}}

Pic[{21, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {21, 1, 1, 0, 0, 1, 1, 1, 1}

Pic[x]

ms21 = { 21, Sum[x[[j]], {j, 2, 9}], ms[x]}

{21, 6, {-0.426336, 0.6003}}

Pic[{22, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {22, 0, 1, 1, 0, 1, 1, 0, 1}

Pic[x]

ms22 = { 22, Sum[x[[j]], {j, 2, 9}], ms[x]}

{22, 5, {-2.21929, 0.520275}}

Pic[{23, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {23, 1, 1, 1, 1, 1, 1, 1, 0, 1}

Pic[**x**]

ms23 = { 23, Sum[**x**[**j**]], {**j**, 2, 9}}, **ms**[**x**]

{23, 7, {-1.41067, 1.40572}}

Pic[{24, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {24, 0, 1, 1, 1, 1, 1, 1, 1}

Pic[**x**]

ms24 = { 24, Sum[x[[j]], {j, 2, 9}], ms[x]}

{24, 7, {-0.802474, 1.04316}}

Pic[{25, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {25, 0, 1, 1, 1, 1, 1, 1, 1}

Pic[x]

ms25 = { 25, Sum[x[[j]], {j, 2, 9}], ms[x]}

{25, 7, {-1.39739, 1.1396}}

Pic[{26, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {26, 1, 1, 1, 1, 1, 1, 1, 1}

Pic2[**x**]

ms26 = { 26, Sum[x[[j]], {j, 2, 9}], ms[x]}

{26, 8, {-0.0888257, 1.88382}}

Pic[{27, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {27, 1, 0, 0, 0, 1, 1, 1, 1}

Pic2[**x**]

ms27 = { 27, Sum[x[[j]], {j, 2, 9}], ms[x]}

{27, 5, {-1.13957, 0.560367}}

Pic[{28, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {28, 1, 1, 0, 0, 1, 1, 1, 1}

Pic[x]

ms28 = { 28, Sum[x[[j]], {j, 2, 9}], ms[x]}

{28, 6, {-0.703732, 0.661624}}

Pic[{29, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {29, 0, 1, 0, 0, 1, 1, 1, 1}

Pic[**x**]

ms29 = { 29, Sum[**x**[**j**]], {**j**, 2, 9}], **ms**[**x**]

{29, 5, {-0.431602, 0.312247}}

Pic[{30, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {30, 1, 1, 1, 1, 1, 1, 1, 1}

Pic[x]

ms30 = { 30, Sum[x[[j]], {j, 2, 9}], ms[x]}

{30, 8, {-0.710913, 0.774848}}

Pic[{31, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {31, 1, 1, 1, 0, 1, 1, 1, 1}

Pic[**x**]

ms31 = { 31, Sum[**x**[[j]], {j, 2, 9}], ms[**x**]}

{31, 7, {-1.0432, 0.427329}}

Pic[{32, 1, 1, 1, 1, 1, 1, 1, 1}]

x = {32, 1, 1, 1, 1, 1, 1, 0, 1}

Pic[**x**]

ms32 = { 32, Sum[**x**[[j]], {j, 2, 9}], ms[**x**]}

{32, 7, {-0.341317, 0.711442}}

```
Pic[{33, 1, 1, 1, 1, 1, 1, 1, 1}]
```

```
x = {33, 1, 1, 1, 1, 1, 0, 1, 1}
```

```
Pic[x]
```

```
ms33 = {33, Sum[x[[j]], {j, 2, 9}], ms[x]}
```

```
{33, 7, {0.172985, 1.05404}}
```

We collect together the mean asymmetric rates for each marker

```
mss = {ms1, ms2, ms3, ms4, ms5, ms6, ms7, ms8, ms9, ms10, ms11, ms12, ms13, ms14, ms15, ms16, ms17,  
ms18, ms19, ms20, ms21, ms22, ms23, ms24, ms25, ms26, ms27, ms28, ms29, ms30, ms31, ms32, ms33};
```

```
MZ = Table[{mss[[j, 1]], mss[[j, 3, 1]], mss[[j, 3, 2]]}, {j, 1, 33}];
```

MatrixForm[MZ]

1	-0.584705	0.39606
2	-0.904069	0.372291
3	1.59396	1.23297
4	-3.23015	0.26646
5	0.0771924	0.45636
6	0.328683	0.577084
7	-0.711104	0.996814
8	1.12315	0.871503
9	-0.131182	1.8411
10	-0.0269243	0.611666
11	1.22549	0.92333
12	-0.0139712	1.55518
13	-0.507136	0.547536
14	-1.21075	1.04285
15	-4.33772	0.468839
16	-1.15513	0.838969
17	1.17759	0.829979
18	-1.20072	0.195669
19	-1.60081	1.09465
20	-1.08809	1.01628
21	-0.426336	0.6003
22	-2.21929	0.520275
23	-1.41067	1.40572
24	-0.802474	1.04316
25	-1.39739	1.1396
26	-0.0888257	1.88382
27	-1.13957	0.560367
28	-0.703732	0.661624
29	-0.431602	0.312247
30	-0.710913	0.774848
31	-1.0432	0.427329
32	-0.341317	0.711442
33	0.172985	1.05404

LB = Table[{ZZ1[[j]] + MZ[[j, 2]], MZ[[j, 3]]}, {j, 1, 33}];

B = Table[{j, Exp[LB[[j, 1]]], LB[[j, 2]]*Exp[LB[[j, 1]]]}, {j, 1, 33}];

Thus the asymmetric rates are

A = Table[{j, B[[j, 2]]/(1 + B[[j, 2]]), B[[j, 3]]/(1 + B[[j, 2]]^2), {j, 1, 33}]; MatrixForm[A]

i.e. the asymmetric constants (and SD) are

A =	1	0.6745914608277626`	0.08694232120498067`
	2	0.4633463923194976`	0.09257247592051233`
	3	0.9732429155540429`	0.03210806731916896`
	4	0.029005874713712673`	0.007504732619953859`
	5	0.6986605998989859`	0.09607929351304731`
	6	0.8204509474124598`	0.08501099550968096`
	7	0.36962272579754907`	0.23225932151607045`
	8	0.9095608279666191`	0.07168980200986297`
	9	0.7342893319987556`	0.3592143789100898`
	10	0.7793282986568104`	0.10519171804607536`
	11	0.9541405214613279`	0.04040159342004441`
	12	0.7027506677978216`	0.324864137462508`
	13	0.5123675151828064`	0.13680022562157804`
	14	0.13942843122652832`	0.1251291716934138`
	15	0.003012211149311151`	0.0014079884245269245`
	16	0.27724063205232263`	0.1681111704826231`
	17	0.9624408662066506`	0.030002459548768477`
	18	0.1535778862226848`	0.025435319456792817`
	19	0.09003122818864748`	0.08968025427835938`
	20	0.2162236572064707`	0.17223041445171394`
	21	0.5181231787952857`	0.1498778201043168`
	22	0.10698102436459021`	0.04970501476507513`
	23	0.1695877971324342`	0.19796499082775937`
	24	0.19452455589105372`	0.1634469119340402`
	25	0.18997530592440287`	0.1753663072102212`
	26	0.670560655538126`	0.4161540139941387`
	27	0.2434904118943821`	0.10322120592000128`
	28	0.38731398742561307`	0.15700463039879492`
	29	0.4475198867563232`	0.07720166353650969`
	30	0.36971174696953457`	0.18055883975068912`
	31	0.25842754010671165`	0.08189448404551788`
	32	0.6028285845146566`	0.17033792642407947`
	33	0.7146170608926542`	0.2149602710368213`

We calc by another method and got the following, note they are within the SD

MatrixForm[A2]

$$\begin{pmatrix} 0.721612 & 0.141106 \\ 0.419695 & 0.142761 \\ 0.886979 & 0.0857226 \\ 0.0647784 & 0.021271 \\ 0.79683 & 0.11209 \\ 0.773628 & 0.0485972 \\ 0.521645 & 0.152886 \\ 0.922729 & 0.0977528 \\ 0.597957 & 0.204874 \\ 0.753922 & 0.131758 \\ 0.854855 & 0.0755958 \\ 0.816615 & 0.0952925 \\ 0.638957 & 0.0839991 \\ 0.349942 & 0.143559 \\ 0.0478602 & 0.0409673 \\ 0.671478 & 0.203361 \\ 0.91627 & 0.10484 \\ 0.342209 & 0.0929699 \\ 0.294443 & 0.133767 \\ 0.488904 & 0.0992638 \\ 0.627817 & 0.130707 \\ 0.460081 & 0.290564 \\ 0.589241 & 0.194343 \\ 0.326405 & 0.190218 \\ 0.259858 & 0.122952 \\ 0.260394 & 0.0518776 \\ 0.281045 & 0.0944989 \\ 0.52146 & 0.124763 \\ 0.592131 & 0.0851968 \\ 0.571669 & 0.0871498 \\ 0.421749 & 0.0846809 \\ 0.478353 & 0.049517 \\ 0.640299 & 0.102388 \end{pmatrix}$$

MatrixForm[Table[{j, A[[j, 2]], A[[j, 3]], A2[[j, 1]], A2[[j, 2]]}, {j, 1, 33}]]

1	0.674591	0.0869423	0.721612	0.141106
2	0.463346	0.0925725	0.419695	0.142761
3	0.973243	0.0321081	0.886979	0.0857226
4	0.0290059	0.00750473	0.0647784	0.021271
5	0.698661	0.0960793	0.79683	0.11209
6	0.820451	0.085011	0.773628	0.0485972
7	0.369623	0.232259	0.521645	0.152886
8	0.909561	0.0716898	0.922729	0.0977528
9	0.734289	0.359214	0.597957	0.204874
10	0.779328	0.105192	0.753922	0.131758
11	0.954141	0.0404016	0.854855	0.0755958
12	0.702751	0.324864	0.816615	0.0952925
13	0.512368	0.1368	0.638957	0.0839991
14	0.139428	0.125129	0.349942	0.143559
15	0.00301221	0.00140799	0.0478602	0.0409673
16	0.277241	0.168111	0.671478	0.203361
17	0.962441	0.0300025	0.91627	0.10484
18	0.153578	0.0254353	0.342209	0.0929699
19	0.0900312	0.0896803	0.294443	0.133767
20	0.216224	0.17223	0.488904	0.0992638
21	0.518123	0.149878	0.627817	0.130707
22	0.106981	0.049705	0.460081	0.290564
23	0.169588	0.197965	0.589241	0.194343
24	0.194525	0.163447	0.326405	0.190218
25	0.189975	0.175366	0.259858	0.122952
26	0.670561	0.416154	0.260394	0.0518776
27	0.24349	0.103221	0.281045	0.0944989
28	0.387314	0.157005	0.52146	0.124763
29	0.44752	0.0772017	0.592131	0.0851968
30	0.369712	0.180559	0.571669	0.0871498
31	0.258428	0.0818945	0.421749	0.0846809
32	0.602829	0.170338	0.478353	0.049517
33	0.714617	0.21496	0.640299	0.102388

```
G = Table[{j, Abs[A[[j, 2]] - 0.5], A[[j, 3]], UnitStep[Abs[A[[j, 2]] - 0.5] - 2*A[[j, 3]]}], {j, 1, 33}];
MatrixForm[G]
```

1	0.174591	0.0869423	1
2	0.0366536	0.0925725	0
3	0.473243	0.0321081	1
4	0.470994	0.00750473	1
5	0.198661	0.0960793	1
6	0.320451	0.085011	1
7	0.130377	0.232259	0
8	0.409561	0.0716898	1
9	0.234289	0.359214	0
10	0.279328	0.105192	1
11	0.454141	0.0404016	1
12	0.202751	0.324864	0
13	0.0123675	0.1368	0
14	0.360572	0.125129	1
15	0.496988	0.00140799	1
16	0.222759	0.168111	0
17	0.462441	0.0300025	1
18	0.346422	0.0254353	1
19	0.409969	0.0896803	1
20	0.283776	0.17223	0
21	0.0181232	0.149878	0
22	0.393019	0.049705	1
23	0.330412	0.197965	0
24	0.305475	0.163447	0
25	0.310025	0.175366	0
26	0.170561	0.416154	0
27	0.25651	0.103221	1
28	0.112686	0.157005	0
29	0.0524801	0.0772017	0
30	0.130288	0.180559	0
31	0.241572	0.0818945	1
32	0.102829	0.170338	0
33	0.214617	0.21496	0

```
Mean[G]
```

```
0.26118
```

Thus there is significant difference from complete symmetric, i.e 16/33 are more than 2 SD from 0.5

SI 5 : Mathematica worksheet for computing mutation rates.

We compute the mutation rates using SNP G2a3, R1b1a2, R1a1, I1, L21, U106, J2, P312. First we download this data for these SNP

55;

The asymmetric rates are entered. We only use the asymmetric rates and its log ratio LB

```
α0 = {{0.6745914608277626, 0.08694232120498067},
{0.4633463923194976, 0.09257247592051233}, {0.9732429155540429, 0.03210806731916896},
{0.029005874713712673, 0.007504732619953859}, {0.6986605998989859, 0.09607929351304731},
{0.8204509474124598, 0.08501099550968096}, {0.36962272579754907, 0.23225932151607045},
{0.9095608279666191, 0.07168980200986297}, {0.7342893319987556, 0.3592143789100898},
{0.7793282986568104, 0.10519171804607536}, {0.9541405214613279, 0.04040159342004441},
{0.7027506677978216, 0.324864137462508}, {0.5123675151828064, 0.13680022562157804},
{0.13942843122652832, 0.1251291716934138}, {0.003012211149311151, 0.0014079884245269245},
{0.27724063205232263, 0.1681111704826231}, {0.9624408662066506, 0.030002459548768477},
{0.1535778862226848, 0.025435319456792817}, {0.09003122818864748, 0.08968025427835938},
{0.2162236572064707, 0.17223041445171394}, {0.5181231787952857, 0.1498778201043168},
{0.10698102436459021, 0.04970501476507513}, {0.1695877971324342, 0.19796499082775937},
{0.19452455589105372, 0.1634469119340402}, {0.18997530592440287, 0.1753663072102212},
{0.670560655538126, 0.4161540139941387}, {0.2434904118943821, 0.10322120592000128},
{0.38731398742561307, 0.15700463039879492}, {0.4475198867563232, 0.07720166353650969},
{0.36971174696953457, 0.18055883975068912}, {0.25842754010671165, 0.08189448404551788},
{0.6028285845146566, 0.17033792642407947}, {0.7146170608926542, 0.2149602710368213}};

LB = {{0.729025827901993, 0.3960604222913862},
{-0.14687791357807545, 0.3722905738010174}, {3.5938344299209577, 1.2329745844759596},
{-3.510822032298842, 0.26646038718036874}, {0.8409278687844486, 0.45636006064334855},
{1.5194056809037075, 0.577084438291061}, {-0.5338356626633742, 0.996813565549179},
{2.308284383892832, 0.871503338048881}, {1.0164951337181467, 1.8411005286954736},
{1.2617563103447185, 0.6116661661273677}, {3.035229051087664, 0.9233302017219662},
{0.8604308710679793, 1.5551762547292591}, {0.04948015338909639, 0.547535896844417},
{-1.8200453515319506, 1.0428461324485363}, {-5.802064111820329, 0.468839110811709},
{-0.9581905054957993, 0.8389690934227843}, {3.243556034437963, 0.829979251701657},
{-1.7068103471817775, 0.19566876748475165}, {-2.3132536922107123, 1.0946547545301615},
{-1.2878103805965928, 1.0162825934127382}, {0.07252448716876952, 0.600299535589346},
{-2.1219563536146264, 0.5202747723446721}, {-1.5885514375725887, 1.4057240423363127},
{-1.4208743104174564, 1.0431577339602012}, {-1.450170638819988, 1.139595552212786},
{0.7107219084237044, 1.8838249946048182}, {-1.1336376419013978, 0.5603670977368852},
{-0.4586168892212833, 0.6616240964076923}, {-0.21069646220858051, 0.3122465649991398},
{-0.5334536194697443, 0.7748476005859001}, {-1.054157531906251, 0.42732889968139737},
{0.41726492065504456, 0.7114420557932557}, {0.9179148956742922, 1.0540393250692588}};
```

We begin with using 33 markers:

```
B = Table[j, {j, 1, 33}]; UU[x_] := 1 - UnitStep[-x];
B = UU[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33}]; BB[j_] := B[[j]];
```

Enter hyperbolic Bessell functions with accuracy

```
n0 = 60; τ = .01; I0[t_] := N[1 + Sum[(t/2)^(2*k) / (k!*(k!)), {k, 1, n0}], n0]; I1[t_] := N[t/2 + Sum[(t/2)^(2*k+1) / (k!*((k+1)!)), {k, 1, n0}], n0];
I2[t_] := N[t^2/8 + Sum[(t/2)^(2*k+2) / (k!*((k+2)!)), {k, 1, n0}], n0]; EE[t_] := N[1 + Sum[t^k / (k!), {k, 1, n0}], n0];
PP = Table[{N[CDF[NormalDistribution[0, 1], -4 + 0.01*j]], -4 + 0.01*j}, {j, 0, 800}];
P = Interpolation[PP]; nn = 500
```

500

We can use reduction to remove singular lineages before computing mutation rates

```

ZZ = Flatten[Table[ {ClearAll[δ0, n01, AAA, AA, VV]; δ0 = δδ[[q1]]; n01 = Count[ Flatten[δ0] , _?Positive] / 33;
  AAA = Flatten[Table[ {ClearAll[L1, δ, m0, f1, m1, m2, m3, m4, m5, m6, m, f2, f3, mm,
    δ1, m10, m11, m12, m13, m14, m16, m17, m18, n, R, RR, β, μ, α, H, TS, TSS];
    n = IntegerPart[n01*.5]; L1 = RandomSample[Range[n01], n]; δ = Table[ δ0[[L1[[j]]] ] , {j, 1, n}];
    m0 = Mean[δ]; m1 = Table[Commonest[ Table[ δ[[k, j]] , {k, 1, n}]][[1]] , {j, 1, 33}];
    m2 = Table[ m1[[j]] , {i, n}, {j, 33}]; m3 = δ - m2; m4 = Abs[m3]; f2[u_] := UnitStep[u - 1];
    m5 = f2[m4]; f3[j_] := N[Sum[ m5[[i, j]] , {i, 1, n}]/n]; m6 = Table[ f3[j] , {j, 1, 33}];
    m = Table[ Sum[m5[[i, j]]*BB[j]*BB0[j], {j, 33}], {i, n}]; mm = N[Sum[m[[i]], {i, n}]/n];
    f1[k_] := 1 - UnitStep[Abs[k] - 0.5]; δ1 = Transpose[δ]; f2[j_, k_] :=
      N[Sum[ f1[δ1[[j, i]] - k] , {i, 1, n}]/n]; R = RandomReal[ {.001, .999}, 33]; RR = P[R];
    β = Table[ Exp[LB[[j, 1]] + LB[[j, 2]]*RR[[j]]] , {j, 1, 33}];

    α = Table[  $\frac{\beta[[j]]}{1 + \beta[[j]]}$  , {j, 1, 33}];

    m10 = Parallelize[Table[ { Max[.001, f2[j, m1[[j]] - 2]], Max[.001, f2[j, m1[[j]] - 1]], Max[.001,
      f2[j, m1[[j]]] ], Max[.001, f2[j, m1[[j]] + 1]], Max[.001, f2[j, m1[[j]] + 2]]}, {j, 1, 33}];
    m11 = Parallelize[Table[ {Min[.999, Max[.001, β[[j]]*m10[[j, 2]]] ,
      Min[.999, Max[.001, m10[[j, 4]]/β[[j]]] } , {j, 1, 33}];
    m12 = Parallelize[Table[ { m10[[j, 4]] - m11[[j, 1]] , m10[[j, 2]] - m11[[j, 2]]}, {j, 1, 33}];
    m13 = Parallelize[Table[ { UnitStep[m12[[j, 1]] - .001] , UnitStep[m12[[j, 2]] - .001] }, {j, 1, 33}];
    m14 = Parallelize[Table[ m13[[j, 1]] - m13[[j, 2]] , {j, 1, 33}];
    m15 = Parallelize[
      Table[ { m10[[j, 2]] - m13[[j, 2]]*m11[[j, 2]] , m10[[j, 3]] - m13[[j, 1]]*m10[[j, 5]]/β[[j]] -
        m13[[j, 2]]*m10[[j, 1]]*β[[j]] , m10[[j, 4]] - m13[[j, 1]]*m11[[j, 1]]}, {j, 1, 33}];
    m16 = Parallelize[Table[ {Min[.999, Max[.001, m15[[j, 1]]] , Min[1, Max[.001, m15[[j, 2]]] ] ,
      Min[.999, Max[.001, m15[[j, 3]]] ] } , {j, 1, 33}];
    m17 = Parallelize[Table[  $\frac{m16[[j, 1]] + m16[[j, 3]]}{m16[[j, 2]] + m16[[j, 1]] + m16[[j, 3]]}$  , {j, 1, 33}];

    { Table[ {k, m17[[k]], 1 - m10[[k, 3]], m14[[k]], α[[k]]} , {k, 1, 33}]} , {q2, 1, nn} ] , 1];

  AA[k_, j_] := Mean[1.0*Table[ AAA[[q2, 1, k, j]] , {q2, 1, nn}]];
  VV[k_, j_] := (1.0*Variance[Table[ AAA[[q2, 1, k, j]] , {q2, 1, nn}]] ) ^ .5;
  {Table[ {k, AA[k, j], VV[k, j]} , {j, 2, 5}, {k, 1, 33}]} , {q1, 1, 8} ] , 2]

```

So the weighted frequencies are

```

ZZZ = Table[ (UnitStep[Abs[ZZ[[q1, 3, k, 2]]] - ZZ[[q1, 3, k, 3]])*ZZ[[q1, 1, k]] +
  (1 - UnitStep[Abs[ZZ[[q1, 3, k, 2]]] - ZZ[[q1, 3, k, 3]])*ZZ[[q1, 2, k]] , {k, 1, 33}, {q1, 1, 8} ]
{{1, 0.241978, 0.0164441}, {1, 0.377293, 0.0269425},
 {1, 0.0607622, 0.00695663}, {1, 0.117902, 0.0058837}, {1, 0.0731556, 0.00799641},
 {1, 0.0836527, 0.00715901}, {1, 0.0996071, 0.00962747}, {1, 0.107988, 0.00955801}},
 {{2, 0.104943, 0.0167327}, {2, 0.3266, 0.0223365}, {2, 0.188417, 0.0315253}, {2, 0.366353, 0.0100219},
 {2, 0.174192, 0.0261706}, {2, 0.418194, 0.0351648}, {2, 0.352017, 0.0520237}, {2, 0.278318, 0.0146238}},
 {{3, 0.132008, 0.0631579}, {3, 0.255496, 0.272028}, {3, 0.797004, 0.182546}, {3, 0.212393, 0.00757349},
 {3, 0.110175, 0.00991895}, {3, 0.0873055, 0.00724728}, {3, 0.454394, 0.0141087}, {3, 0.180202, 0.0157082}},
 {{4, 0.102079, 0.0102725}, {4, 0.423653, 0.0903945}, {4, 0.586556, 0.222162}, {4, 0.0953325, 0.00600721},
 {4, 0.447135, 0.028128}, {4, 0.378465, 0.0272033}, {4, 0.22902, 0.0154296}, {4, 0.40879, 0.0613807}},
 {{5, 0.361158, 0.128869}, {5, 0.155706, 0.035591}, {5, 0.129199, 0.0169759}, {5, 0.349396, 0.0405065},
 {5, 0.130066, 0.0103975}, {5, 0.10318, 0.00776532}, {5, 0.493168, 0.0144349}, {5, 0.174536, 0.0123476}},
 {{6, 0.303032, 0.0807956}, {6, 0.413887, 0.0240941}, {6, 0.197114, 0.0367884}, {6, 0.313228, 0.00849702},
 {6, 0.335432, 0.0141003}, {6, 0.193523, 0.0366218}, {6, 0.882996, 0.145808}, {6, 0.251299, 0.0521692}},
 {{7, 0.00581639, 0.00214944}, {7, 0.175887, 0.0181814}, {7, 0.016989, 0.00373541},
 {7, 0.00441132, 0.00122069}, {7, 0.022642, 0.00474759}, {7, 0.0157232, 0.00324917},
 {7, 0.0111839, 0.00298299}, {7, 0.00709173, 0.00260596}}, {{8, 0.383006, 0.0169197},
 {8, 0.0406508, 0.0107346}, {8, 0.0676882, 0.00679814}, {8, 0.10294, 0.00596202}, {8, 0.0111099, 0.00473071},
 {8, 0.0229164, 0.00368855}, {8, 0.506845, 0.255313}, {8, 0.0167379, 0.00665848}},
 {{9, 0.375984, 0.0140796}, {9, 0.367, 0.0225886}, {9, 0.301024, 0.139547}, {9, 0.225533, 0.0074581},
 {9, 0.412763, 0.0155406}, {9, 0.404781, 0.0129465}, {9, 0.541594, 0.0144435}, {9, 0.397707, 0.0149789}},
 {{10, 0.0697695, 0.0212009}, {10, 0.173395, 0.0275832}, {10, 0.231694, 0.0120304},
 {10, 0.0781648, 0.0132444}, {10, 0.171619, 0.0118778},
 {10, 0.153984, 0.00912639}, {10, 0.396341, 0.0351642}, {10, 0.225753, 0.0319193}},
 {{11, 0.0331453, 0.00558373}, {11, 0.225435, 0.0194828}, {11, 0.0275307, 0.00435882},
 {11, 0.0342678, 0.00344625}, {11, 0.0864942, 0.00921059},
 {11, 0.055588, 0.0121441}, {11, 0.0474903, 0.00640516}, {11, 0.0458951, 0.0116272}},
 {{12, 0.350216, 0.013934}, {12, 0.3606, 0.0230288}, {12, 0.394721, 0.0136318}, {12, 0.180733, 0.0067671},
 {12, 0.279152, 0.0141652}, {12, 0.244042, 0.0110227}, {12, 0.435606, 0.0142797}, {12, 0.20988, 0.0133126}},
 {{13, 0.482506, 0.133079}, {13, 0.547044, 0.122554}, {13, 0.360596, 0.0844214}, {13, 0.382715, 0.0092507},

```

```
{13, 0.486206, 0.0155756}, {13, 0.514888, 0.0133398}, {13, 0.718516, 0.0109366}, {13, 0.489307, 0.015902}},
{{14, 0.0631047, 0.0127401}, {14, 0.0540097, 0.0132292}, {14, 0.0474961, 0.00562182},
{14, 0.0606609, 0.00840915}, {14, 0.032392, 0.00681518},
{14, 0.0206105, 0.00417113}, {14, 0.463357, 0.0170255}, {14, 0.0461488, 0.00891647}},
{{15, 0.127621, 0.193965}, {15, 0.388544, 0.169484}, {15, 0.341923, 0.155075}, {15, 0.0660717, 0.114187},
{15, 0.293508, 0.163879}, {15, 0.273477, 0.157338}, {15, 0.139817, 0.107128}, {15, 0.377025, 0.16259}},
{{16, 0.0328, 0.00523059}, {16, 0.0232957, 0.00742916}, {16, 0.019345, 0.0044359},
{16, 0.00973868, 0.00177264}, {16, 0.0253301, 0.0056628},
{16, 0.0242559, 0.00392986}, {16, 0.0447129, 0.00578726}, {16, 0.016919, 0.00466428}},
{{17, 0.0564918, 0.00670366}, {17, 0.0132609, 0.0053507}, {17, 0.0186331, 0.00405953},
{17, 0.0120693, 0.00220126}, {17, 0.00959922, 0.0033265},
{17, 0.0125625, 0.0030954}, {17, 0.129623, 0.00976858}, {17, 0.0234378, 0.00638117}},
{{18, 0.254765, 0.0138628}, {18, 0.293332, 0.0296257}, {18, 0.527212, 0.135086}, {18, 0.320387, 0.0124398},
{18, 0.2835, 0.0170622}, {18, 0.480489, 0.0242167}, {18, 0.921044, 0.112732}, {18, 0.31685, 0.0206874}},
{{19, 0.110554, 0.00889138}, {19, 0.135573, 0.0500814}, {19, 0.00762689, 0.00244561},
{19, 0.0635473, 0.00432753}, {19, 0.102342, 0.0179717}, {19, 0.128547, 0.0226461}, {19, 0.46064, 0.060156},
{19, 0.284503, 0.0139433}}, {{20, 0.326187, 0.0136383}, {20, 0.121128, 0.0211237},
{20, 0.142485, 0.00978087}, {20, 0.0961906, 0.0154173}, {20, 0.123015, 0.0232792},
{20, 0.0591964, 0.0106955}, {20, 0.726567, 0.180892}, {20, 0.306091, 0.0144728}},
{{21, 0.703089, 0.0127418}, {21, 0.6678, 0.0215936}, {21, 0.623635, 0.0133578}, {21, 0.522201, 0.129189},
{21, 0.631911, 0.0102436}, {21, 0.579679, 0.0128866}, {21, 0.744435, 0.010262}, {21, 0.439514, 0.11945}},
{{22, 0.568914, 0.168208}, {22, 0.284082, 0.0295834}, {22, 0.254853, 0.015754}, {22, 0.27212, 0.0181003},
{22, 0.19818, 0.0423121}, {22, 0.239382, 0.0471068}, {22, 0.644455, 0.138678}, {22, 0.307509, 0.0142398}},
{{23, 0.38838, 0.0142677}, {23, 0.480619, 0.19386}, {23, 0.595135, 0.218288}, {23, 0.150443, 0.00647607},
{23, 0.288463, 0.0142537}, {23, 0.366587, 0.0122914}, {23, 0.515977, 0.214796}, {23, 0.346049, 0.0154354}},
{{24, 0.218029, 0.0169974}, {24, 0.0699043, 0.0119252}, {24, 0.0377197, 0.0054793},
{24, 0.0575882, 0.155748}, {24, 0.0171655, 0.00483844}, {24, 0.0490679, 0.00548332},
{24, 0.0580067, 0.0913623}, {24, 0.02925, 0.00620488}}, {{25, 0.125137, 0.0183529},
{25, 0.110896, 0.0139663}, {25, 0.239493, 0.0117175}, {25, 0.0547481, 0.00427558},
{25, 0.168663, 0.023705}, {25, 0.15895, 0.0088793}, {25, 0.355716, 0.0134139}, {25, 0.117637, 0.01054}},
{{26, 0.297459, 0.0132852}, {26, 0.54333, 0.0230184}, {26, 0.550296, 0.0130874}, {26, 0.288316, 0.00823433},
{26, 0.567428, 0.0108124}, {26, 0.607285, 0.0116625}, {26, 0.513432, 0.0146813}, {26, 0.563451, 0.188722}},
{{27, 0.28247, 0.0569122}, {27, 0.61098, 0.161511}, {27, 0.38708, 0.0140864}, {27, 0.164581, 0.0161486},
{27, 0.188307, 0.0281477}, {27, 0.208144, 0.0102606}, {27, 0.36964, 0.0693049}, {27, 0.163161, 0.0255089}},
{{28, 0.704574, 0.0132327}, {28, 0.637322, 0.0200894}, {28, 0.611817, 0.0134795}, {28, 0.487891, 0.14152},
{28, 0.557467, 0.0148684}, {28, 0.586793, 0.179292}, {28, 0.715458, 0.0100426}, {28, 0.60052, 0.0159204}},
{{29, 0.437626, 0.0572555}, {29, 0.353487, 0.0528997}, {29, 0.606995, 0.0141607}, {29, 0.75519, 0.138632},
{29, 0.298839, 0.0439394}, {29, 0.390819, 0.0517461}, {29, 0.692616, 0.0134485}, {29, 0.38197, 0.0531262}},
{{30, 0.698066, 0.0127693}, {30, 0.715278, 0.0191426}, {30, 0.700488, 0.217916}, {30, 0.590805, 0.00945024},
{30, 0.767015, 0.199283}, {30, 0.663376, 0.0113416}, {30, 0.897519, 0.143493}, {30, 0.661171, 0.0148794}},
{{31, 0.965891, 0.0775317}, {31, 0.810255, 0.183083}, {31, 0.975611, 0.0638617}, {31, 0.853859, 0.142067},
{31, 0.870208, 0.151199}, {31, 0.8562, 0.139299}, {31, 0.878421, 0.130794}, {31, 0.811993, 0.169577}},
{{32, 0.151053, 0.0312768}, {32, 0.258817, 0.0212773}, {32, 0.692992, 0.207514}, {32, 0.230117, 0.00762533},
{32, 0.233918, 0.0137314}, {32, 0.328982, 0.0119735}, {32, 0.446036, 0.084218}, {32, 0.247142, 0.0141075}},
{{33, 0.0515174, 0.0140446}, {33, 0.0863217, 0.0131092}, {33, 0.0598047, 0.00644727},
{33, 0.032726, 0.00328817}, {33, 0.041856, 0.00601193}, {33, 0.0650372, 0.0117121},
{33, 0.080729, 0.00779351}, {33, 0.0489113, 0.00668937}}}
```

We begin by assuming all mutation rates = .002 with zero error

```
μ0 = Table[ {.002, 0}, {j, 1, 33}]
```

```
{{.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0},
{.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0},
{.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0},
{.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}, {.002, 0}}
```

We use hyperbolic BesselI function to compute expansion times for each marker

```

μ = Table[ {μ0[[j, 1]]*α0[[j, 1]], μ0[[j, 1]]*(1 - α0[[j, 1]])}, {j, 1, 33}];

H[t_, j_] := EE[-t*μ0[[j, 1]]] *  $\left( \text{I0}\left[2\,t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right] + \left(\sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}}\right) * \right.$ 
 $\text{I1}\left[2*\tau*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right] * \left(\frac{\mu[[j, 2]]}{\mu[[j, 1]]}\right) * \text{I2}\left[2*(1 - \tau)*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right]$ 
 $+ \left(\sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}}\right) * \text{I1}\left[2\,t*\tau*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right] * \left(\frac{\mu[[j, 1]]}{\mu[[j, 2]]}\right) *$ 
 $\left. \text{I2}\left[2*(1 - \tau)*t*\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right] \right)$ ;

TS[k_] := Normal[InverseSeries[Series[ (1 - H[x, k]), {x, 0, 60}]]];
TSS[t_, k_] := TS[k] /. x -> t;
ZZZ01 = Parallelize[Table[ TSS[ZZZ[[k, q1, 2]], k], {k, 1, 33}, {q1, 1, 8}]];
ZZZ02 = Parallelize[
  Table[ 0.5*(TSS[ZZZ[[k, q1, 2]] + ZZZ[[k, q1, 3]], k) - TSS[ZZZ[[k, q1, 2]] - ZZZ[[k, q1, 3]], k) ],
    {k, 1, 33}, {q1, 1, 8}]];

```

Thes etiems for each marker and SNP are

ZZZ01

```

{{147.621, 267.798, 31.9008, 64.6248, 38.6708, 44.6596, 53.2013, 59.0156},
{58.1379, 222.661, 112.315, 260.293, 101.842, 323.413, 246.816, 178.763},
{72.8783, 135.691, 816.584, 120.546, 58.0865, 46.0076, 309.186, 105.241},
{53.5192, 277.822, 471.523, 50.3016, 301.87, 239.905, 130.731, 266.733},
{262.455, 90.9927, 71.4175, 233.934, 71.5963, 56.028, 407.047, 99.8266},
{193.902, 289.974, 113.242, 199.062, 218.161, 110.744, 1700.23, 148.946},
{2.9124, 101.287, 8.62295, 2.24825, 11.2896, 7.98964, 5.77438, 3.69428},
{251.23, 21.2728, 35.4194, 54.5678, 5.75, 11.4408, 331.397, 8.54033},
{263.068, 254.719, 185.168, 134.668, 303.588, 291.528, 474.621, 286.857},
{36.2827, 98.0227, 84.1058, 41.101, 97.1904, 85.9118, 279.438, 134.678},
{16.8692, 130.119, 14.0437, 17.5119, 45.0292, 29.1102, 24.3455, 23.3463},
{239.779, 249.68, 284.951, 103.953, 176.018, 148.629, 329.794, 123.455},
{414.145, 517.197, 253.658, 279.195, 413.695, 465.843, 1181.92, 422.267},
{32.8306, 28.0877, 24.8358, 31.2358, 16.7416, 10.2021, 340.459, 23.7743},
{86.6778, 258.359, 213.041, 30.9218, 165.992, 158.326, 73.523, 241.588},
{16.7347, 12.4356, 9.88897, 4.92628, 13.0532, 9.34431, 23.5914, 8.75359},
{29.1714, 6.69125, 9.55869, 6.00314, 4.8461, 6.23107, 70.1033, 11.8729},
{153.625, 182.571, 431.754, 204.315, 174.95, 362.095, 2009.75, 200.578},
{59.368, 74.1536, 3.72892, 32.8145, 53.0071, 68.5922, 328.613, 171.045},
{211.32, 66.2606, 79.5521, 52.1405, 66.7289, 30.7049, 954.401, 194.747},
{1081.16, 883.278, 724.306, 508.108, 751.487, 602.563, 1397.63, 331.882},
{438.981, 170.798, 151.475, 163.455, 113.978, 161.881, 592.843, 191.137},
{264.788, 383.819, 539.598, 83.7403, 179.675, 245.292, 419.081, 225.999},
{127.148, 37.1673, 19.2294, 34.1457, 8.67046, 25.196, 38.8963, 14.9101},
{70.7709, 60.3665, 143.533, 28.314, 95.562, 88.7622, 237.421, 64.0221},
{193.243, 495.053, 508.184, 185.771, 540.239, 630.802, 443.84, 547.701},
{181.902, 630.823, 271.804, 91.8968, 108.734, 122.177, 254.208, 92.4832},
{1032., 744.788, 665.084, 416.975, 537.92, 604.562, 1084.03, 639.381},
{341.027, 252.396, 670.56, 1519.14, 192.226, 287.839, 1001.01, 275.036},
{970.586, 1062.97, 997.534, 607.619, 1384.71, 824.093, 4314.41, 818.591},
{7527.64, 1519.36, 8339.69, 1979.47, 2193.29, 1946.23, 2216.36, 1603.6},
{85.3533, 162.553, 965.123, 140.701, 143.873, 223.73, 350.871, 152.545},
{27.2471, 46.0164, 31.1939, 16.7035, 21.2089, 33.5272, 26.9997, 24.8934}}

```

with SD

ZZZ02

```
{12.8735, 27.1544, 3.54967, 3.6819, 4.47719, 3.94605, 5.92509, 5.86692},
{10.4934, 20.7229, 23.0994, 11.1357, 17.5376, 45.1309, 56.4437, 11.8036},
{45.0736, 178.946, 741.223, 5.03162, 5.69807, 3.92592, 12.5211, 114.123},
{5.58782, 85.8037, 321.128, 3.25797, 24.1807, 22.3423, 9.5966, 51.1488},
{166.911, 21.8166, 9.80429, 41.8157, 6.11008, 4.52236, 21.0067, 8.14247},
{63.2607, 23.2247, 24.5126, 6.86846, 12.8299, 29.1906, 5648.88, 38.3098},
{1.12512, 12.0561, 2.07912, 0.600412, 2.28334, 1.66283, 1.52765, 1.30863},
{14.2765, 6.02288, 3.99538, 3.0569, 2.38594, 1.98518, 263.515, 3.46258},
{13.8188, 23.9295, 92.5119, 5.52936, 18.4096, 13.1232, 22.9388, 16.7724},
{11.6827, 17.5026, 29.3367, 7.71569, 7.1209, 5.87839, 34.6872, 22.2411},
{2.99458, 12.7839, 2.38938, 1.83049, 4.94405, 14.0901, 3.05739, 6.19263},
{14.0011, 20.4719, 14.2816, 4.60196, 11.2167, 8.02171, 16.5945, 9.50148},
{235.452, 273.418, 87.2182, 10.4269, 24.6675, 23.1883, 75.0802, 25.7094},
{6.54708, 7.17845, 3.14899, 4.58536, 3.50967, 2.08725, 17.2307, 5.03282},
{148.594, 156.998, 130.102, 46.1505, 99.2634, 99.7638, 55.3584, 147.25},
{2.70565, 3.74262, 2.39156, 0.948874, 3.06906, 2.231, 3.15643, 2.21167},
{3.34882, 2.64986, 1.89491, 1.15916, 1.53505, 1.69221, 5.608, 3.27624},
{10.0398, 23.6468, 19.1483, 10.429, 13.8877, 22.2969, 4550., 17.1849},
{5.268, 27.7755, 1.20086, 2.53413, 10.8197, 14.5214, 67.9157, 11.1398},
{11.2818, 12.2546, 6.22569, 9.81319, 13.6833, 5.52836, 1029.11, 12.1147},
{78.545, 102.534, 43.763, 293.655, 36.7723, 31.3213, 109.972, 130.708},
{213.566, 18.5004, 10.4592, 11.9131, 26.2584, 8.62417, 299.493, 11.9307},
{13.5516, 294.036, 503.384, 4.02208, 11.3142, 11.6027, 342.037, 13.2627},
{11.5215, 6.53521, 2.75294, 94.5326, 2.28239, 3.05312, 83.0723, 3.40538},
{34.7476, 8.58825, 9.15956, 2.26757, 17.4859, 5.94649, 11.5307, 5.87584},
{11.1531, 40.0835, 26.8587, 6.83926, 20.9877, 30.9891, 23.425, 514.502},
{45.0757, 475.431, 13.1905, 10.8558, 19.2094, 7.23041, 63.387, 17.0187},
{68.0499, 67.2083, 37.7732, 228.988, 31.2534, 550.409, 60.7848, 42.5118},
{79.3371, 50.6448, 39.8183, 3027.99, 36.9248, 59.2341, 71.1488, 59.189},
{64.8322, 106.092, 3034.24, 20.2762, 14879.6, 39.6927, 1.73364×106, 56.7734},
{69879., 8919.09, 65067.4, 7115.19, 21671.5, 9366.33, 13811.8, 8132.49},
{20.5094, 16.1882, 2855.41, 5.88682, 10.1813, 11.1055, 119.615, 10.9515},
{7.62156, 7.08381, 3.53107, 1.73704, 3.25475, 6.59028, 10.0363, 3.74717}}
```

We begin by assuming all mutation rates = .002 with zero error

```
ZZZ1 = Log[ZZZ01]; ZZZ2 = Log[ZZZ02 + ZZZ01] - Log[ZZZ01];
SZZ = Table[Sort[Table[{BB[j]*ZZZ1[[j, q]], ZZZ2[[j, q]]}, {j, 1, 33}]], {q, 1, 8}]; ZZO = Table[
Sum[SZZ[[q, j, 1]]/SZZ[[q, j, 2]]^2, {j, 15, 25}]/Sum[1/SZZ[[q, j, 2]]^2, {j, 15, 25}], {q, 1, 8}];
ZZZ3 = Parallelize[Table[{ZZZ1[[j, q1]] - ZZO[[q1]], ZZZ2[[j, q1]]}, {j, 1, 33}, {q1, 1, 8}]; ZZZ31 =
Parallelize[Table[Sort[Table[{ZZZ1[[j, q1]] - ZZO[[q1]], ZZZ2[[j, q1]]}, {q1, 1, 8}]], {j, 1, 33}];
ZZZ32 = Table[Sort[Table[{ZZZ31[[j, k + 3, 1]] - ZZZ31[[j, k, 1]],
Sum[ZZZ31[[j, q, 1]]/4, {q, k, 3 + k}]], {k, 1, 5}]], {j, 1, 33}];
ZZ1 = Parallelize[Table[{j, 0.5*(ZZZ32[[j, 1, 2]] + ZZZ32[[j, 2, 2]]),
0.5*(ZZZ32[[j, 1, 1]] + ZZZ32[[j, 2, 1]])}, {j, 1, 33}]]
{{1, -1.48387, 0.737843}, {2, 0.0107028, 0.559742}, {3, -0.581556, 0.478216}, {4, 0.368936, 0.336573},
{5, -0.99344, 0.59345}, {6, 0.134912, 0.482254}, {7, -4.04085, 0.516394}, {8, -2.74896, 0.962118},
{9, 0.223648, 0.258596}, {10, -0.75845, 0.459867}, {11, -2.24489, 0.68006}, {12, -0.103645, 0.208006},
{13, 0.811537, 0.125636}, {14, -2.14669, 0.414502}, {15, -0.0155295, 0.432367},
{16, -3.07451, 0.316299}, {17, -3.4185, 0.248648}, {18, 0.0690082, 0.352519}, {19, -1.1786, 0.280496},
{20, -1.21791, 0.633725}, {21, 1.316, 0.229851}, {22, -0.129738, 0.416643}, {23, 0.176497, 0.170971},
{24, -2.39774, 0.655777}, {25, -0.764317, 0.301404}, {26, 0.644058, 0.530232},
{27, -0.457427, 0.174324}, {28, 1.17814, 0.0808416}, {29, 0.320102, 0.327311},
{30, 1.49961, 0.0637986}, {31, 2.23122, 0.563484}, {32, -0.150163, 0.230759}, {33, -2.07363, 0.154063}}
```

We compute mean log time of middle entries. We bootstrap to estimate CI


```

ZZZ100 = Log[ZZZ01]; ZZZ200 = Log[ ZZZ02 + ZZZ01] - Log[ ZZZ01];
BBB = Flatten[Table[ {ClearAll[R, i]; W = Table[1, {j, 1, 33}]; R = RandomReal[ {.01, .99}, 33]; RR = P[R];
ZZZ11 = Table[ Exp[ZZZ100[[j]] + ZZZ200[[j]]*RR[[j]]], {j, 1, 33}];
ZZZ12 = ZZZ02; i = 1; While[ i < 11, {ZZZ1 = Log[ZZZ11]; ZZZ2 = Log[ZZZ11 + ZZZ12] - Log[ZZZ11];
SZZ = Table[ Sort[ Table[ {BB[j]*ZZZ1[[j, q]], ZZZ2[[j, q]]}, {j, 1, 33}], {q, 1, 8}];
ZZ0 = Table[ Sum[ SZZ[[q, j, 1]]/SZZ[[q, j, 2]]^2, {j, 15, 25}]/
Sum[ 1/SZZ[[q, j, 2]]^2, {j, 15, 25}], {q, 1, 8}]; ZZZ3 = Parallelize[
Table[ {ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {j, 1, 33}, {q1, 1, 8}]; ZZZ31 = Parallelize[
Table[ Sort[ Table[ {ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {q1, 1, 8}], {j, 1, 33}];
ZZZ32 = Table[ Sort[ Table[ { ZZZ31[[j, k + 3, 1]] - ZZZ31[[j, k, 1]],
Sum[ ZZZ31[[j, q, 1]]/4, {q, k, 3 + k}], {k, 1, 5}], {j, 1, 33}];
ZZ1 = Parallelize[Table[ 0.5*(ZZZ32[[j, 1, 2]] + ZZZ32[[j, 2, 2]]), {j, 1, 33}];
ZZ2 = Parallelize[Table[ 0.5*(ZZZ32[[j, 1, 1]] + ZZZ32[[j, 2, 1]]), {j, 1, 33}];
a1 = Sum[ ZZ1[[j]], {j, 1, 33}]/33; a2 = Exp[a1]; ZZ3 = a2*Exp[-ZZ1]; ZZ4 = a2*Exp[-ZZ1]*ZZ2;
ZZZ03 = Table[ ZZZ11[[j, q1]]*ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ11 = ZZZ03;
ZZZ04 = Table[ ZZZ12[[j, q1]]*ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ12 = ZZZ04;
ZZ5 = Table[ W[[j]]*ZZ3[[j]], {j, 1, 33}]; W = ZZ5; i++; {W}}, {k1, 1, 250}, 2];

BBB1 = Mean[BBB]; BBB2 = (Variance[BBB])^0.5;
WW = Table[ {Re[BBB1[[j]]], BBB2[[j]]}, {j, 1, 33}]; MatrixForm[WW]

```

2.28512	0.323262
0.600462	0.0943589
1.25972	0.673063
0.36297	0.0746295
1.38816	0.189061
0.53932	0.198732
32.4926	8.78557
8.13256	1.78686
0.464922	0.0597547
1.04262	0.10844
5.26481	1.30458
0.571891	0.0600025
0.223576	0.0221806
4.44163	0.897095
0.564392	0.245787
10.8708	2.65751
15.8351	3.65597
0.478321	0.08286
1.78692	0.380184
1.8072	0.258491
0.149356	0.0213048
0.591291	0.0778801
0.440946	0.0478114
6.06255	1.63236
1.09445	0.144306
0.251905	0.0828199
0.855531	0.104782
0.173622	0.0281207
0.378542	0.0833605
0.121505	0.0109229
0.19028	0.334059
0.637694	0.0738936
4.36244	0.647695

This produces the first iterate for mutation rates

```

μ00 = Table[ {μ0[[j, 1]]/WW[[j, 1]], WW[[j, 2]]*μ0[[j, 1]]/WW[[j, 1]]^2}, {j, 1, 33}]

{{0.000875229, 0.000123813}, {0.00333077, 0.000523411}, {0.00158765, 0.000848272}, {0.0055101, 0.00113292},
{0.00144076, 0.000196224}, {0.00370838, 0.00136649}, {0.0000615525, 0.000016643},
{0.000245925, 0.0000540337}, {0.0043018, 0.000552895}, {0.00191825, 0.000199512},
{0.00037988, 0.0000941317}, {0.00349717, 0.000366921}, {0.00894551, 0.00088747}, {0.000450285, 0.000090946},
{0.00354364, 0.00154322}, {0.00018398, 0.0000449765}, {0.000126302, 0.0000291602},
{0.00418129, 0.000724329}, {0.00111925, 0.00023813}, {0.00110668, 0.000158293}, {0.0133909, 0.00191014},
{0.00338243, 0.000445506}, {0.0045357, 0.000491802}, {0.000329894, 0.000088825}, {0.0018274, 0.000240946},
{0.00793951, 0.00261031}, {0.00233773, 0.000286314}, {0.0115192, 0.00186571}, {0.00528343, 0.00116349},
{0.0164602, 0.00147971}, {0.0105109, 0.0184531}, {0.0031363, 0.000363422}, {0.000458459, 0.0000680679}}

```

Now we repeat with the new μ0

```
μ0 = μ00;
```

```
μ = Table[ {μ0[[j, 1]]*α0[[j, 1]], μ0[[j, 1]]*(1 - α0[[j, 1]])}, {j, 1, 33}];
```

$$\begin{aligned}
 H[t_, j_] := & EE[-t * \mu_0[[j, 1]]] * \left(I_0 \left[2 t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]} \right] + \left(\sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}} \right) * \right. \\
 & I_1 \left[2 * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]} \right] * \left(\frac{\mu[[j, 2]]}{\mu[[j, 1]]} \right) * I_2 \left[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]} \right] \\
 & + \left(\sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}} \right) * I_1 \left[2 t * \tau * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]} \right] * \left(\frac{\mu[[j, 1]]}{\mu[[j, 2]]} \right) * \\
 & \left. I_2 \left[2 * (1 - \tau) * t * \sqrt{\mu[[j, 1]] * \mu[[j, 2]]} \right] \right);
 \end{aligned}$$

```
TS[k_] := Normal[InverseSeries[Series[ (1 - H[x, k]), {x, 0, 60}]]];
```

```
TSS[t_, k_] := TS[k] /. x -> t;
```

```
ZZZ01 = Parallelize[Table[ TSS[ZZZ[[k, q1, 2]], k], {k, 1, 33}, {q1, 1, 8}]];
```

```
ZZZ02 = Parallelize[
```

```
Table[ 0.5 * (TSS[ZZZ[[k, q1, 2]] + ZZZ[[k, q1, 3]], k) - TSS[ZZZ[[k, q1, 2]] - ZZZ[[k, q1, 3]], k),
{k, 1, 33}, {q1, 1, 8}]];
```

zzz01

```
{337.332, 611.95, 72.897, 147.675, 88.3672, 102.052, 121.571, 134.858},
{34.9096, 133.699, 67.441, 156.296, 61.1519, 194.197, 148.204, 107.34},
{91.8064, 170.933, 1028.67, 151.855, 73.173, 57.9569, 389.489, 132.574},
{19.4259, 100.841, 171.149, 18.258, 109.57, 87.0785, 47.4514, 96.816},
{364.329, 126.312, 99.1388, 324.737, 99.3871, 77.7758, 565.046, 138.575},
{104.575, 156.389, 61.0737, 107.358, 117.659, 59.7262, 916.966, 80.3293},
{94.6312, 3291.07, 280.182, 73.0513, 366.829, 259.604, 187.624, 120.037},
{2043.14, 173.003, 288.05, 443.776, 46.7622, 93.0428, 2695.11, 69.4548},
{122.306, 118.424, 86.0884, 62.6101, 141.145, 135.538, 220.662, 133.366},
{37.829, 102.2, 87.6902, 42.8526, 101.332, 89.5731, 291.347, 140.417},
{88.8134, 685.054, 73.9373, 92.197, 237.07, 153.26, 128.175, 122.914},
{137.128, 142.79, 162.961, 59.4501, 100.663, 84.9996, 188.606, 70.6026},
{92.5929, 115.633, 56.7118, 62.4212, 92.4921, 104.151, 264.248, 94.4086},
{145.822, 124.755, 110.311, 138.738, 74.3598, 45.3141, 1512.19, 105.596},
{48.9203, 145.816, 120.239, 17.452, 93.6846, 89.3579, 41.4958, 136.351},
{181.919, 135.184, 107.501, 53.5524, 141.898, 101.58, 256.456, 95.1581},
{461.932, 105.957, 151.363, 95.0604, 76.7385, 98.6697, 1110.09, 188.009},
{73.4823, 87.3278, 206.517, 97.7283, 83.6824, 173.198, 961.306, 95.9407},
{106.086, 132.506, 6.66327, 58.6368, 94.7193, 122.569, 587.205, 305.644},
{381.898, 119.746, 143.767, 94.2284, 120.593, 55.49, 1724.79, 351.947},
{161.477, 131.922, 108.179, 75.8887, 112.239, 89.9962, 208.744, 49.5683},
{259.565, 100.992, 89.566, 96.6496, 67.3943, 95.7186, 350.543, 113.018},
{116.757, 169.243, 237.934, 36.925, 79.2268, 108.16, 184.792, 99.6535},
{770.839, 225.329, 116.579, 207.01, 52.5651, 152.752, 235.811, 90.3933},
{77.4555, 66.0683, 157.091, 30.9884, 104.588, 97.1461, 259.846, 70.0692},
{48.6788, 124.706, 128.014, 46.7965, 136.089, 158.902, 111.805, 137.968},
{155.623, 539.689, 232.537, 78.6206, 93.0251, 104.526, 217.483, 79.1223},
{179.179, 129.312, 115.474, 72.3963, 93.3949, 104.966, 188.212, 111.011},
{129.093, 95.5426, 253.835, 575.059, 72.7658, 108.959, 378.925, 104.113},
{117.932, 129.157, 121.206, 73.8291, 168.25, 100.132, 524.224, 99.4633},
{1432.36, 289.103, 1586.87, 376.652, 417.338, 370.327, 421.728, 305.133},
{54.4293, 103.659, 615.454, 89.7241, 91.7468, 142.671, 223.748, 97.2768},
{118.864, 200.744, 136.082, 72.8679, 92.5226, 146.26, 117.785, 108.596}
```

ZZZ02

```
{ {29.4174, 62.0509, 8.11141, 8.41356, 10.2309, 9.01718, 13.5395, 13.4066},
{6.3009, 12.4433, 13.8703, 6.68656, 10.5307, 27.0994, 33.8923, 7.08758},
{56.7803, 225.422, 933.736, 6.33845, 7.178, 4.94558, 15.7731, 143.763},
{2.02821, 31.1441, 116.56, 1.18255, 8.77688, 8.10958, 3.48328, 18.5655},
{231.699, 30.2849, 13.6099, 58.0469, 8.48176, 6.27776, 29.1606, 11.303},
{34.1178, 12.5255, 13.2201, 3.70429, 6.91944, 15.7431, 3046.55, 20.6612},
{36.5581, 391.733, 67.5558, 19.5089, 74.1917, 54.0296, 49.6374, 42.5206},
{116.104, 48.9815, 32.4927, 24.8605, 19.4038, 16.1446, 2143.05, 28.1596},
{6.42466, 11.1253, 43.0108, 2.57072, 8.55902, 6.10126, 10.6647, 7.79785},
{12.1806, 18.2485, 30.5869, 8.04451, 7.42437, 6.12891, 36.1655, 23.1889},
{15.7659, 67.305, 12.5797, 9.63717, 26.0295, 74.1818, 16.0966, 32.603},
{8.00708, 11.7077, 8.1675, 2.63182, 6.41472, 4.58754, 9.49024, 5.43381},
{52.6414, 61.1296, 19.4999, 2.3312, 5.51505, 5.18434, 16.7861, 5.74801},
{29.0797, 31.884, 13.9866, 20.3665, 15.5887, 9.27077, 76.5325, 22.3539},
{83.865, 88.6086, 73.4287, 26.047, 56.0235, 56.3059, 31.2438, 83.1067},
{29.4124, 40.6851, 25.9981, 10.315, 33.363, 24.2527, 34.3127, 24.0425},
{53.0288, 41.9608, 30.0061, 18.3553, 24.3076, 26.7963, 88.8033, 51.8795},
{4.80224, 11.3108, 9.15905, 4.98841, 6.6428, 10.6651, 2176.36, 8.21989},
{9.41348, 49.6326, 2.14584, 4.52829, 19.3339, 25.9486, 121.36, 19.9058},
{20.3884, 22.1465, 11.2511, 17.7344, 24.7285, 9.99085, 1859.8, 21.8936},
{11.7311, 15.314, 6.53625, 43.859, 5.49215, 4.678, 16.4249, 19.522},
{126.28, 10.9391, 6.18443, 7.0441, 15.5264, 5.0994, 177.087, 7.05454},
{5.97553, 129.654, 221.965, 1.77352, 4.98894, 5.11619, 150.82, 5.84813},
{69.8496, 39.62, 16.6898, 573.109, 13.8371, 18.5097, 503.63, 20.6453},
{38.0296, 9.39944, 10.0247, 2.48175, 19.1375, 6.50816, 12.6199, 6.43084},
{2.80952, 10.0972, 6.76583, 1.72284, 5.2869, 7.80629, 5.90087, 129.606},
{38.5637, 406.746, 11.2849, 9.28751, 16.4343, 6.18584, 54.2296, 14.56},
{11.815, 11.6689, 6.55828, 39.7574, 5.4263, 95.5634, 10.5536, 7.381},
{30.0324, 19.1712, 15.0729, 1146.22, 13.9776, 22.4226, 26.9328, 22.4055},
{7.87747, 12.8908, 368.677, 2.46366, 1807.95, 4.82288, 210.647, 6.89828},
{13.296.5, 1697.12, 12.381, 1353.87, 4123.65, 1782.22, 2628.1, 1547.45},
{13.0787, 10.3231, 1820.88, 3.75399, 6.49258, 7.08194, 76.2778, 6.98374},
{33.2486, 30.9027, 15.4041, 7.57771, 14.1986, 28.7497, 43.7827, 16.3468} }
```

```
ID[q_] := IntegerDigits[q, 2, 3]
```

```
Q[x_, q_] := Sum[Q[x, q, j] * bb[[q, j]], {j, 1, 33}]
```

```
Pic2[q_] := Plot[Q[x, q], {x, 1, 800}, PlotRange -> {0, .2},
PlotStyle -> RGBColor[ID[q][[1]], 0.5 * ID[q][[2]], ID[q][[3]]],
PlotLabel -> {q}]
```

```
ZZZ100 = Log[ZZZ01]; ZZZ200 = Log[ZZZ02 + ZZZ01] - Log[ZZZ01];
BBB = Flatten[Table[{ClearAll[R, i]; W = Table[1, {j, 1, 33}]; R = RandomReal[{.01, .99}, 33]; RR = P[R];
ZZZ11 = Table[Exp[ZZZ100[[j]] + ZZZ200[[j]] * RR[[j]]], {j, 1, 33}];
ZZZ12 = ZZZ02; i = 1; While[i < 11, {ZZZ1 = Log[ZZZ11]; ZZZ2 = Log[ZZZ11 + ZZZ12] - Log[ZZZ11];
SZZ = Table[Sort[Table[{BB[j] * ZZZ1[[j, q]], ZZZ2[[j, q]]], {j, 1, 33}], {q, 1, 8}];
ZZ0 = Table[Sum[SZZ[[q, j, 1]] / SZZ[[q, j, 2]] ^ 2, {j, 15, 25}] /
Sum[1 / SZZ[[q, j, 2]] ^ 2, {j, 15, 25}], {q, 1, 8}]; ZZZ3 = Parallelize[
Table[{ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {j, 1, 33}, {q1, 1, 8}]; ZZZ31 = Parallelize[
Table[Sort[Table[{ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {q1, 1, 8}], {j, 1, 33}];
ZZZ32 = Table[Sort[Table[{ZZZ31[[j, k + 3, 1]] - ZZZ31[[j, k, 1]],
Sum[ZZZ31[[j, q, 1]] / 4, {q, k, 3 + k}], {k, 1, 5}], {j, 1, 33}];
ZZ1 = Parallelize[Table[0.5 * (ZZZ32[[j, 1, 2]] + ZZZ32[[j, 2, 2]]), {j, 1, 33}];
ZZ2 = Parallelize[Table[0.5 * (ZZZ32[[j, 1, 1]] + ZZZ32[[j, 2, 1]]), {j, 1, 33}];
a1 = Sum[ZZ1[[j]], {j, 1, 33}] / 33; a2 = Exp[a1]; ZZ3 = a2 * Exp[-ZZ1]; ZZ4 = a2 * Exp[-ZZ1] * ZZ2;
ZZZ03 = Table[ZZZ11[[j, q1]] * ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ11 = ZZZ03;
ZZZ04 = Table[ZZZ12[[j, q1]] * ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ12 = ZZZ04;
ZZ5 = Table[W[[j]] * ZZ3[[j]], {j, 1, 33}; W = ZZ5; i++; {W}], {k1, 1, 250}], 2];
```

```
BBB1 = Mean[BBB]; BBB2 = (Variance[BBB]) ^ .5;
WW = Table[{Re[BBB1[[j]]], BBB2[[j]]}, {j, 1, 33}]; MatrixForm[WW]
```

```
( 1.04024  0.167468
  1.07642  0.189871
  1.0672   0.58683
  1.069    0.224411
  1.05327  0.151853
  1.05798  0.400839
  1.02185  0.26036
  1.08672  0.260047
  1.08653  0.119089
  1.05523  0.108724
  1.06295  0.269837
  1.05986  0.113711
  1.04598  0.112418
  1.09441  0.221301
  1.12627  0.500083
  1.06997  0.231586
  1.04799  0.240816
  1.05108  0.182342
  1.0694   0.211226
  1.03015  0.157624
  1.06853  0.156392
  1.07839  0.13664
  1.07373  0.13874
  1.04476  0.283189
  1.06566  0.147162
  1.02603  0.305433
  1.06048  0.138716
  1.04561  0.164493
  1.07191  0.222104
  1.05398  0.0903946
  0.913448  1.6529
  1.04447  0.119725
  1.06303  0.153357)
```

```
μ00 = Table[{μ0[[j, 1]]/WW[[j, 1]], WW[[j, 2]]*μ0[[j, 1]]/WW[[j, 1]]^2}, {j, 1, 33}]
```

Thus we get the second iterate, more have negliable difference

```
μ00 = {{0.0008413745642801459, 0.00013545284477059174},
{0.0030943065004157364, 0.0005458090834901315}, {0.001487683024607187, 0.000818046938144343},
{0.005154423776176086, 0.0010820421485960823}, {0.0013678883678863688, 0.00019721226307880675},
{0.00350515201541656, 0.0013280045110582817}, {0.00006023615058638974, 0.000015347676838542014},
{0.00022630091094626249, 0.00005415295868208447}, {0.0039592202164502694, 0.00043394994438830935},
{0.001817850343110373, 0.0001872991323636844}, {0.00035738481257634647, 0.00009072485735000124},
{0.0032996407048695537, 0.0003540139931716074}, {0.008552282984829827, 0.0009191697432282981},
{0.00041143978374449865, 0.00008319712152116025}, {0.0031463405314982895, 0.0013970253750291844},
{0.00017194892516357016, 0.000037216941765983825}, {0.00012051809931757855, 0.00002769374130740949},
{0.003978085016968495, 0.0006901204586930073}, {0.0010466147270345773, 0.00020672678541880626},
{0.0010742945444384023, 0.00016437878694035347}, {0.012532096258901716, 0.0018342318580006015},
{0.003136560577007424, 0.0003974254631384725}, {0.004224243568063607, 0.0005458257532284053},
{0.00031576180697374235, 0.00008558971548308784}, {0.0017147939648363615, 0.0002368023501967496},
{0.007738103585059785, 0.0023035185785761593}, {0.0022044104733743744, 0.00028834886084192716},
{0.011016725520319137, 0.0017331208436325459}, {0.00492900194442053, 0.0010213144251913927},
{0.01561715540816209, 0.0013394066043832702}, {0.011506784189978763, 0.02082170328776407},
{0.0030027577371241373, 0.0003441977673199861}, {0.000431277134456353, 0.00006221794891128538}}
```

We now recalc omitting the markers with error > 33%

```
B = Table[j, {j, 1, 33}]; UU[x_] := 1 - UnitStep[-x];
B = UU[{1, 2, 0, 4, 5, 0, 7, 8, 9, 10, 11, 12, 13, 14, 0, 16, 17,
  18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 0, 32, 33}]; BB[j_] := B[[j]];
```

```

μ0 = {{0.0008413745642801459`, 0.00013545284477059174`,
{0.0030943065004157364`, 0.0005458090834901315`, {0.001487683024607187`, 0.000818046938144343`,
{0.005154423776176086`, 0.0010820421485960823`, {0.0013678883678863688`, 0.00019721226307880675`,
{0.00350515201541656`, 0.0013280045110582817`, {0.00006023615058638974`, 0.000015347676838542014`,
{0.00022630091094626249`, 0.00005415295868208447`, {0.0039592202164502694`, 0.00043394994438830935`,
{0.001817850343110373`, 0.0001872991323636844`, {0.00035738481257634647`, 0.00009072485735000124`,
{0.0032996407048695537`, 0.0003540139931716074`, {0.008552282984829827`, 0.0009191697432282981`,
{0.00041143978374449865`, 0.00008319712152116025`, {0.0031463405314982895`, 0.0013970253750291844`,
{0.00017194892516357016`, 0.000037216941765983825`, {0.00012051809931757855`, 0.00002769374130740949`,
{0.003978085016968495`, 0.0006901204586930073`, {0.0010466147270345773`, 0.00020672678541880626`,
{0.0010742945444384023`, 0.00016437878694035347`, {0.012532096258901716`, 0.0018342318580006015`,
{0.003136560577007424`, 0.0003974254631384725`, {0.004224243568063607`, 0.0005458257532284053`,
{0.00031576180697374235`, 0.00008558971548308784`, {0.0017147939648363615`, 0.0002368023501967496`,
{0.007738103585059785`, 0.0023035185785761593`, {0.0022044104733743744`, 0.00028834886084192716`,
{0.011016725520319137`, 0.0017331208436325459`, {0.00492900194442053`, 0.0010213144251913927`,
{0.01561715540816209`, 0.0013394066043832702`, {0.011506784189978763`, 0.02082170328776407`,
{0.0030027577371241373`, 0.0003441977673199861`, {0.000431277134456353`, 0.00006221794891128538`}};

μ = Table[ {μ0[[j, 1]]*α0[[j, 1]], μ0[[j, 1]]*(1 - α0[[j, 1]])}, {j, 1, 33}];

H[t_, j_] := EE[-t*μ0[[j, 1]]]*

$$I0\left[2t\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right] + \left(\sqrt{\frac{\mu[[j, 1]]}{\mu[[j, 2]]}}\right)*$$


$$I1\left[2t\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right]*\left(\frac{\mu[[j, 2]]}{\mu[[j, 1]]}\right)*I2\left[2*(1 - t)*t\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right]$$


$$+ \left(\sqrt{\frac{\mu[[j, 2]]}{\mu[[j, 1]]}}\right)*I1\left[2t\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right]*\left(\frac{\mu[[j, 1]]}{\mu[[j, 2]]}\right)*$$


$$I2\left[2*(1 - t)*t\sqrt{\mu[[j, 1]]*\mu[[j, 2]]}\right];$$

TS[k_] := Normal[InverseSeries[Series[(1 - H[x, k]), {x, 0, 60}]]];
TSS[t_, k_] := TS[k] /. x -> t;
ZZZ01 = Parallelize[Table[TSS[ZZZ[[k, q1, 2]], k], {k, 1, 33}, {q1, 1, 8}]];
ZZZ02 = Parallelize[
Table[0.5*(TSS[ZZZ[[k, q1, 2]] + ZZZ[[k, q1, 3]], k) - TSS[ZZZ[[k, q1, 2]] - ZZZ[[k, q1, 3]], k)],
{k, 1, 33}, {q1, 1, 8}]];

```

zzz01

```
{352.095, 636.957, 75.5595, 153.452, 91.8506, 105.901, 127.718, 139.409},
{36.8756, 143.469, 71.3815, 169.159, 65.1099, 207.429, 159.57, 115.635},
{95.5167, 199.868, 1119.97, 161.5, 78.7048, 61.5538, 413.889, 134.26},
{20.9532, 108.621, 175.837, 19.4924, 116.964, 93.534, 50.8356, 103.524},
{365.692, 128.485, 104.265, 348.989, 105.04, 81.5259, 596.157, 146.388},
{109.123, 166.614, 64.8009, 113.869, 124.547, 63.4375, 994.835, 86.4174},
{96.9737, 3370.73, 285.608, 73.4715, 382.259, 264.079, 187.205, 118.348},
{2225.84, 184.015, 311.518, 484.405, 49.4136, 102.64, 3328.48, 74.6927},
{132.533, 128.038, 97.8072, 68.1311, 152.057, 147.658, 239.801, 143.837},
{40.2925, 108.428, 152.218, 45.417, 107.161, 94.7944, 306.776, 147.544},
{94.4553, 722.963, 78.2097, 97.7155, 254.142, 160.434, 136.433, 131.731},
{145.013, 151.115, 172.351, 63.1604, 107.064, 90.4031, 200.552, 75.2926},
{96.0302, 123.143, 59.8407, 65.399, 97.3915, 108.675, 276.106, 98.548},
{159.687, 135.859, 118.969, 153.257, 80.3505, 50.7441, 1645.02, 115.493},
{43.4116, 156.577, 133.158, 21.7299, 110.544, 101.639, 47.8899, 150.627},
{195.267, 137.737, 114.055, 57.0267, 149.985, 143.513, 268.513, 99.5794},
{483.521, 110.822, 156.173, 100.799, 80.0623, 104.946, 1157.77, 196.96},
{76.9783, 91.6052, 210.891, 102.505, 87.7789, 181.416, 1033.87, 101.048},
{113.034, 140.902, 7.31973, 63.0732, 104.087, 132.982, 622.97, 329.15},
{395.874, 122.938, 147.02, 95.8139, 125.034, 57.4008, 1654.23, 364.18},
{171.735, 142.146, 115.45, 76.2616, 119.818, 95.7566, 223.16, 55.6334},
{293.728, 110.18, 96.5818, 104.535, 71.9702, 89.6439, 369.598, 121.576},
{125.721, 172.533, 250.055, 39.5256, 84.8351, 116.064, 193.668, 107.368},
{811.402, 232.169, 122.51, 189.62, 54.9838, 160.614, 191.054, 94.4565},
{79.6351, 69.8319, 166.998, 33.1237, 110.969, 103.789, 276.4, 74.4461},
{49.8457, 128.05, 131.378, 47.8449, 140.003, 162.944, 114.799, 137.942},
{161.073, 540.572, 246.467, 84.4689, 98.5855, 110.849, 230.759, 83.6419},
{185.338, 134.685, 120.9, 74.919, 97.3962, 109.237, 196.474, 115.433},
{140.101, 100.69, 271.312, 600.428, 79.7116, 116.934, 406.965, 112.922},
{124.134, 135.633, 125.653, 77.6092, 183.877, 105.276, 624.435, 104.234},
{1135.92, 250.373, 1434.74, 328.718, 369.32, 334.063, 393.027, 252.877},
{56.858, 108.108, 646.361, 93.3345, 95.2391, 148.585, 235.605, 101.987},
{123.992, 213.327, 144.834, 77.6819, 100.021, 158.128, 198.645, 117.492}}
```

ZZZ02

```
{ {29.6161, 66.8078, 9.05617, 8.40383, 10.6148, 9.6638, 13.3384, 13.4273},
{6.40646, 13.6911, 14.1123, 6.84207, 11.3868, 28.2272, 34.3235, 7.94666},
{49.37, 262.622, 1065.45, 6.54583, 7.53926, 5.36323, 17.9619, 131.92},
{2.23321, 31.7051, 124.059, 1.29562, 10.2269, 8.73453, 3.94147, 20.8486},
{192.07, 33.3158, 15.165, 56.8566, 9.30051, 6.64205, 30.7846, 11.9412},
{37.5245, 14.1201, 14.0239, 3.99589, 6.93753, 13.8772, 2957.54, 21.8772},
{35.9904, 404.541, 63.595, 20.397, 81.5182, 55.212, 50.3466, 43.7168},
{132.092, 49.789, 32.6002, 29.9089, 21.1785, 16.7457, 2990.18, 30.008},
{7.12792, 11.1891, 61.0191, 2.71698, 8.66229, 7.06162, 12.1317, 8.01939},
{12.8653, 19.7019, 9.5166, 8.13549, 8.45406, 6.30801, 39.5337, 24.9988},
{16.2076, 72.0275, 12.5726, 10.0158, 28.4397, 36.1644, 18.8969, 34.2422},
{8.08551, 13.7234, 8.87754, 2.74138, 6.97616, 5.04383, 10.449, 5.69578},
{53.0967, 66.682, 20.9145, 2.40148, 5.77955, 5.57932, 18.6807, 5.97487},
{33.5829, 34.4547, 14.5159, 22.0935, 17.2561, 10.4035, 91.723, 22.983},
{71.9367, 90.754, 76.5369, 39.0727, 75.2612, 70.0828, 39.8272, 85.1758},
{31.8806, 44.6616, 26.5154, 10.4516, 34.1426, 23.6572, 35.8973, 27.7839},
{59.2048, 45.0382, 34.3705, 18.5044, 27.8888, 26.0348, 94.0802, 54.3127},
{5.0798, 11.6451, 95.3696, 5.14314, 6.58228, 14.3506, 2391.67, 8.49744},
{9.74059, 56.8093, 2.35761, 4.46371, 19.4801, 25.4163, 119.8, 19.734},
{21.9753, 23.4286, 11.218, 16.4551, 25.8974, 10.809, 1783.13, 22.3546},
{12.614, 15.5536, 6.86929, 42.349, 5.57057, 5.06124, 16.1502, 26.8045},
{162.17, 14.1138, 7.15521, 8.45891, 17.6044, 20.9011, 170.29, 7.07029},
{6.47792, 120.507, 217.609, 1.89359, 5.27203, 5.31969, 153.77, 6.39761},
{74.8472, 41.5636, 18.2545, 541.571, 15.6762, 18.5567, 314.704, 20.4341},
{12.7731, 9.51221, 9.85083, 2.68473, 17.6693, 6.51342, 14.1858, 7.25131},
{2.93858, 10.8648, 6.35099, 1.78357, 5.6573, 7.40729, 6.13624, 123.089},
{41.5146, 363.472, 12.9666, 9.41555, 17.1091, 6.45791, 61.6253, 14.8424},
{12.9335, 11.8065, 6.74916, 43.3434, 5.58752, 106.045, 10.8174, 7.47645},
{30.9297, 21.96, 16.3666, 1140.92, 15.7761, 23.9498, 29.6602, 23.9484},
{7.9482, 13.8661, 435.961, 2.89955, 2985.33, 5.41492, 223.153, 7.0034},
{1115.8, 1180.12, 9047.8, 1277.95, 3836.31, 1255.45, 2195.14, 787.399},
{13.3729, 11.3068, 1790.32, 3.80578, 6.9099, 7.51835, 76.526, 7.30388},
{35.1045, 34.5693, 16.3163, 7.99149, 14.8097, 29.88, 20.3704, 16.653} }
```

```
ZZZ100 = Log[ZZZ01]; ZZZ200 = Log[ ZZZ02 + ZZZ01] - Log[ ZZZ01];
BBB = Flatten[Table[ {ClearAll[R, i]; W = Table[1, {j, 1, 33}]; R = RandomReal[{.01, .99}, 33]; RR = P[R];
ZZZ11 = Table[ Exp[ZZZ100[[j]] + ZZZ200[[j]]*RR[[j]]], {j, 1, 33}];
ZZZ12 = ZZZ02; i = 1; While[ i < 11, {ZZZ1 = Log[ZZZ11]; ZZZ2 = Log[ZZZ11 + ZZZ12] - Log[ZZZ11];
SZZ = Table[ Sort[ Table[ {BB[j]*ZZZ1[[j, q]], ZZZ2[[j, q]]}, {j, 1, 33}], {q, 1, 8}];
ZZ0 = Table[ Sum[ SZZ[[q, j, 1]]/SZZ[[q, j, 2]]^2, {j, 15, 25}]/
Sum[ 1/SZZ[[q, j, 2]]^2, {j, 15, 25}], {q, 1, 8}]; ZZZ3 = Parallelize[
Table[ {ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {j, 1, 33}, {q1, 1, 8}]; ZZZ31 = Parallelize[
Table[Sort[Table[ {ZZZ1[[j, q1]] - ZZ0[[q1]], ZZZ2[[j, q1]]}, {q1, 1, 8}], {j, 1, 33}];
ZZZ32 = Table[ Sort[Table[ {ZZZ31[[j, k + 3, 1]] - ZZZ31[[j, k, 1]],
Sum[ ZZZ31[[j, q, 1]]/4, {q, k, 3 + k}], {k, 1, 5}], {j, 1, 33}];
ZZ1 = Parallelize[Table[ 0.5*(ZZZ32[[j, 1, 2]] + ZZZ32[[j, 2, 2]]), {j, 1, 33}];
ZZ2 = Parallelize[Table[ 0.5*(ZZZ32[[j, 1, 1]] + ZZZ32[[j, 2, 1]]), {j, 1, 33}];
a1 = Sum[ ZZ1[[j]], {j, 1, 33}]/33; a2 = Exp[a1]; ZZ3 = a2*Exp[-ZZ1]; ZZ4 = a2*Exp[-ZZ1]*ZZ2;
ZZZ03 = Table[ ZZZ11[[j, q1]]*ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ11 = ZZZ03;
ZZZ04 = Table[ ZZZ12[[j, q1]]*ZZ3[[j]], {j, 1, 33}, {q1, 1, 8}]; ZZZ12 = ZZZ04;
ZZ5 = Table[ W[[j]]*ZZ3[[j]], {j, 1, 33}]; W = ZZ5; i++; {W}, {k1, 1, 250}, 2];
```



```
BBB1 = Mean[BBB]; BBB2 = (Variance[BBB]) ^ .5;
WW = Table[{Re[BBB1[[j]]], BBB2[[j]]}, {j, 1, 33}]; MatrixForm[WW]
```

1.1649	0.160619
1.22697	0.215462
1.14123	0.589415
1.03562	0.20614
1.07868	0.135385
1.12136	0.380589
0.930481	0.227181
1.05115	0.226821
1.05295	0.119168
0.938514	0.0957675
0.98662	0.26942
1.11483	0.122663
1.07007	0.0908328
1.0335	0.182608
1.05592	0.494792
1.0532	0.216189
1.05371	0.227001
1.04586	0.158481
1.05156	0.189169
0.922971	0.197697
1.07054	0.148951
1.19094	0.159132
1.07444	0.10712
0.984939	0.229655
1.22143	0.214678
0.954924	0.21793
1.02751	0.134041
1.03377	0.11509
1.07086	0.219375
1.06167	0.100681
0.858581	1.72122
1.03444	0.115947
0.995835	0.140488

```
μ00 = Table[{μ0[[j, 1]]/WW[[j, 1]], WW[[j, 2]]*μ0[[j, 1]]/WW[[j, 1]]^2}, {j, 1, 33}]
```

```
μ00 = {{0.0007222714614464581, 0.00009958813591473886},
{0.00252191560852714, 0.00044286260571714243}, {0.001303572925590671, 0.000673257413391668},
{0.004977150289157533, 0.0009907051039416278}, {0.0012681155803029092, 0.00015916137630843307},
{0.0031258069210570354, 0.0010608984520777312}, {0.00006473656291412204, 0.000015805722766883475},
{0.0002152896050608881, 0.00004645613833311312}, {0.00376011308999344, 0.0004255511924985024},
{0.001936945271013519, 0.00019764896844221686}, {0.00036223149405663986, 0.00009891596711384029},
{0.0029597631499781647, 0.00032565834123501734}, {0.00799224117747206, 0.0006784189013479164},
{0.0003981048976878356, 0.00007034111713507639}, {0.0029797070704232153, 0.0013962524295180574},
{0.0001632628725620396, 0.000033512674531545074}, {0.00011437505872541055, 0.000024639866368253074},
{0.0038036479009438554, 0.0005763737033982009}, {0.0009952936025767742, 0.00017904592014773458},
{0.0011639531081141532, 0.00024931403005722065}, {0.011706339153923283, 0.0016287823454677738},
{0.002633694999084507, 0.00035191241088457974}, {0.003931568575211744, 0.00039197138452279784},
{0.0003205903099024094, 0.00007475086059909508}, {0.001403927013011494, 0.00024675398596351206},
{0.008103373405530107, 0.001849329896817691}, {0.0021453934823239366, 0.0002798708167225011},
{0.010656815015340831, 0.001186419087516791}, {0.004602824417133062, 0.0009429258440670032},
{0.014709975691241446, 0.0013949821075148583}, {0.013402087511949615, 0.026867449582444667},
{0.002902798923344392, 0.00032536758376112187}, {0.0004330809176749126, 0.00006109732296106685}};
```

Using all 33 markers we got mutation rates

```

μ2 = {{0.0008413745642801459`, 0.00013545284477059174`,
{0.0030943065004157364`, 0.0005458090834901315`, {0.001487683024607187`, 0.000818046938144343`,
{0.005154423776176086`, 0.0010820421485960823`, {0.0013678883678863688`, 0.00019721226307880675`,
{0.00350515201541656`, 0.0013280045110582817`, {0.00006023615058638974`, 0.000015347676838542014`,
{0.00022630091094626249`, 0.00005415295868208447`, {0.0039592202164502694`, 0.00043394994438830935`,
{0.001817850343110373`, 0.0001872991323636844`, {0.00035738481257634647`, 0.00009072485735000124`,
{0.0032996407048695537`, 0.0003540139931716074`, {0.008552282984829827`, 0.0009191697432282981`,
{0.00041143978374449865`, 0.00008319712152116025`, {0.0031463405314982895`, 0.0013970253750291844`,
{0.00017194892516357016`, 0.000037216941765983825`, {0.00012051809931757855`, 0.00002769374130740949`,
{0.003978085016968495`, 0.0006901204586930073`, {0.0010466147270345773`, 0.00020672678541880626`,
{0.0010742945444384023`, 0.00016437878694035347`, {0.012532096258901716`, 0.0018342318580006015`,
{0.003136560577007424`, 0.0003974254631384725`, {0.004224243568063607`, 0.0005458257532284053`,
{0.00031576180697374235`, 0.00008558971548308784`, {0.0017147939648363615`, 0.0002368023501967496`,
{0.007738103585059785`, 0.0023035185785761593`, {0.0022044104733743744`, 0.00028834886084192716`,
{0.011016725520319137`, 0.0017331208436325459`, {0.00492900194442053`, 0.0010213144251913927`,
{0.01561715540816209`, 0.0013394066043832702`, {0.011506784189978763`, 0.02082170328776407`,
{0.0030027577371241373`, 0.0003441977673199861`, {0.000431277134456353`, 0.00006221794891128538`}};

```

We compare and see these are within the SD

```

MatrixForm[Table[{j, μ00[[j, 1]], μ00[[j, 2]]/μ00[[j, 1]], μ2[[j, 1]], μ2[[j, 2]]/μ2[[j, 1]]}, {j, 1, 33}]]

```

1	0.000722271	0.137882	0.000841375	0.16099
2	0.00252192	0.175606	0.00309431	0.176391
3	0.00130357	0.516471	0.00148768	0.54988
4	0.00497715	0.199051	0.00515442	0.209925
5	0.00126812	0.12551	0.00136789	0.144173
6	0.00312581	0.3394	0.00350515	0.378872
7	0.0000647366	0.244154	0.0000602362	0.254792
8	0.00021529	0.215784	0.000226301	0.239296
9	0.00376011	0.113175	0.00395922	0.109605
10	0.00193695	0.102042	0.00181785	0.103033
11	0.000362231	0.273074	0.000357385	0.253858
12	0.00295976	0.110029	0.00329964	0.107289
13	0.00799224	0.0848847	0.00855228	0.107477
14	0.000398105	0.17669	0.00041144	0.20221
15	0.00297971	0.468587	0.00314634	0.444016
16	0.000163263	0.205268	0.000171949	0.216442
17	0.000114375	0.21543	0.000120518	0.229789
18	0.00380365	0.151532	0.00397809	0.173481
19	0.000995294	0.179893	0.00104661	0.197519
20	0.00116395	0.214196	0.00107429	0.153011
21	0.0117063	0.139137	0.0125321	0.146363
22	0.00263369	0.133619	0.00313656	0.126707
23	0.00393157	0.0996985	0.00422424	0.129213
24	0.00032059	0.233166	0.000315762	0.271058
25	0.00140393	0.17576	0.00171479	0.138094
26	0.00810337	0.228217	0.0077381	0.297685
27	0.00214539	0.130452	0.00220441	0.130805
28	0.0106568	0.111133	0.0110167	0.157317
29	0.00460282	0.204858	0.004929	0.207205
30	0.01471	0.0948324	0.0156172	0.0857651
31	0.0134021	2.00472	0.0115068	1.80952
32	0.0029028	0.112088	0.00300276	0.114627
33	0.000433081	0.141076	0.000431277	0.144264

```
 $\mu_0 = \text{Table}[\{j, \mu_{00}[[j, 1]], \mu_{00}[[j, 2]] / \mu_{00}[[j, 1]]\}, \{j, 1, 33\}]; \text{MatrixForm}[\mu_0]$ 
```

1	0.000722271	0.137882
2	0.00252192	0.175606
3	0.00130357	0.516471
4	0.00497715	0.199051
5	0.00126812	0.12551
6	0.00312581	0.3394
7	0.0000647366	0.244154
8	0.00021529	0.215784
9	0.00376011	0.113175
10	0.00193695	0.102042
11	0.000362231	0.273074
12	0.00295976	0.110029
13	0.00799224	0.0848847
14	0.000398105	0.17669
15	0.00297971	0.468587
16	0.000163263	0.205268
17	0.000114375	0.21543
18	0.00380365	0.151532
19	0.000995294	0.179893
20	0.00116395	0.214196
21	0.0117063	0.139137
22	0.00263369	0.133619
23	0.00393157	0.0996985
24	0.00032059	0.233166
25	0.00140393	0.17576
26	0.00810337	0.228217
27	0.00214539	0.130452
28	0.0106568	0.11133
29	0.00460282	0.204858
30	0.01471	0.0948324
31	0.0134021	2.00472
32	0.0029028	0.112088
33	0.000433081	0.141076

which are the mutation rates and % errors we use (we shall not use $j = 3, 6, 15, 31$ as error > 33%)

SI 6: TMRCA for G2a2, R1b1a2, R1a1a, I1, L21, U106, J2, P312

This file is a summary of the MATHEMATICA worksheet in which the computations were done. Download FTDNA data (33 markers) for G2a2(Western Europe), R1b1a2,R1a1, I1, L21, U106, J2, P312

551;

After the method of reduction is applied we obtained files (q), marker (j), reduced frequency f0, unreduced frequency f0, mean $\pm$ and

SD, MatrixForm[ZZ[[1]]]

1	1	0.271316	0.302637	1.	0.0178978	0.0126611	0.
1	2	0.201682	0.300628	0.476	0.0271493	0.0130464	0.869741
1	3	0.178373	0.116323	-0.96	0.195153	0.00940206	0.280562
1	4	0.385975	0.352135	1.	0.029537	0.0130741	0.
1	5	0.157559	0.223067	0.992	0.0278212	0.0111525	0.126491
1	6	0.263609	0.394063	-0.024	0.0879698	0.0135655	0.989617
1	7	0.141216	0.177767	0.4	0.0227142	0.010168	0.909565
1	8	0.0349038	0.0394798	-0.952	0.00624372	0.00567578	0.279299
1	9	0.394868	0.403013	-0.152	0.1575	0.0127973	0.990363
1	10	0.184642	0.219704	-0.876	0.0224796	0.0107642	0.479106
1	11	0.226154	0.251632	-0.628	0.0615687	0.0110867	0.77201
1	12	0.351466	0.378108	-0.424	0.115257	0.0130506	0.907479
1	13	0.509485	0.587614	0.936	0.0981115	0.014322	0.352706
1	14	0.0633869	0.078296	0.648	0.0132519	0.0072736	0.747214
1	15	0.442594	0.251704	1.	0.164795	0.0110868	0.
1	16	0.0197234	0.0253991	0.256	0.00523943	0.00399652	0.926227
1	17	0.0098425	0.0118655	-0.844	0.00292294	0.00312497	0.395334
1	18	0.291304	0.317184	1.	0.0184938	0.013626	0.
1	19	0.127226	0.135211	0.932	0.0204479	0.00974217	0.357614
1	20	0.0994612	0.1147	0.968	0.0157589	0.00802896	0.251454
1	21	0.587612	0.66217	0.112	0.111483	0.012874	0.995702
1	22	0.322642	0.352448	0.992	0.0211385	0.0139905	0.126491
1	23	0.564598	0.446161	0.944	0.232074	0.0135255	0.330607
1	24	0.0302449	0.0624574	0.104	0.00865472	0.00665072	0.934171
1	25	0.0591171	0.122888	-0.18	0.0195145	0.00921815	0.96713
1	26	0.522952	0.525399	0.852	0.220455	0.0129397	0.52075
1	27	0.730244	0.518251	1.	0.169299	0.0147761	0.
1	28	0.653017	0.661103	0.188	0.163942	0.0107722	0.982097
1	29	0.365704	0.465498	0.984	0.0420302	0.0138983	0.178526
1	30	0.744854	0.682332	0.652	0.170513	0.0129874	0.757092
1	31	0.920049	0.693498	0.936	0.130351	0.0114332	0.352706
1	32	0.170861	0.234063	-0.376	0.0308018	0.0119116	0.919787
1	33	0.0521186	0.0713901	-0.228	0.0115408	0.00636655	0.961099

MatrixForm[ZZ[[2]]]

2	1	0.375804	0.405941	1.	0.0192334	0.0159503	0.
2	2	0.216383	0.32804	0.052	0.0270099	0.0163081	0.990566
2	3	0.24417	0.134482	-0.916	0.246669	0.011885	0.396956
2	4	0.416149	0.3687	0.948	0.0732266	0.0160445	0.312549
2	5	0.157481	0.231802	0.984	0.0292686	0.0142515	0.178526
2	6	0.303033	0.412304	-0.228	0.10148	0.0171384	0.96942
2	7	0.137046	0.175617	0.332	0.0237766	0.0130006	0.943045
2	8	0.0413054	0.045835	-0.9	0.00858105	0.00692682	0.432142
2	9	0.351883	0.366785	-0.308	0.133925	0.017788	0.951186
2	10	0.1726	0.209822	-0.892	0.0232874	0.0124879	0.448487
2	11	0.197252	0.22631	-0.596	0.0540341	0.0150227	0.797073
2	12	0.313909	0.36268	-0.512	0.0794011	0.0161954	0.860709
2	13	0.550854	0.607683	0.896	0.113249	0.0149098	0.444945
2	14	0.0529518	0.0672475	0.668	0.0118638	0.00837816	0.715418
2	15	0.410581	0.242587	0.944	0.165348	0.0143759	0.262946
2	16	0.0178889	0.0238416	-0.468	0.00611965	0.00512932	0.836526
2	17	0.0127908	0.0133597	-0.58	0.00408374	0.00366713	0.623824
2	18	0.295365	0.314046	1.	0.0227443	0.01457	0.
2	19	0.136005	0.144515	0.92	0.0228784	0.0109555	0.392705
2	20	0.120162	0.135855	0.948	0.0198027	0.0116342	0.312549
2	21	0.60818	0.667749	0.28	0.124676	0.0168221	0.957742
2	22	0.281875	0.306323	0.984	0.0221234	0.0153316	0.178526
2	23	0.484618	0.415974	0.952	0.195875	0.0170885	0.306712
2	24	0.0357675	0.0695446	0.3	0.010597	0.00837306	0.941033
2	25	0.0605304	0.110416	-0.288	0.0187234	0.0104575	0.942661
2	26	0.547557	0.543617	0.648	0.21323	0.0159789	0.763168
2	27	0.630696	0.482178	1.	0.167021	0.0178532	0.
2	28	0.592336	0.637967	0.06	0.132286	0.0141712	0.998191
2	29	0.352739	0.453875	0.992	0.0480101	0.016667	0.126491
2	30	0.793362	0.716792	0.52	0.162863	0.0151199	0.85588
2	31	0.830828	0.698904	0.848	0.167285	0.0117022	0.523442
2	32	0.196029	0.258535	-0.708	0.0348402	0.0140828	0.704786
2	33	0.0652856	0.0869967	-0.392	0.0139458	0.00903857	0.913056

MatrixForm[ZZ[[3]]]

3	1	0.0370043	0.0608496	0.608	0.0115158	0.00486699	0.754063
3	2	0.18861	0.283093	-0.856	0.0294776	0.009238	0.518013
3	3	0.803939	0.54632	-1.	0.184399	0.0108528	0.
3	4	0.569174	0.500711	0.912	0.186773	0.00918263	0.390983
3	5	0.129998	0.167628	-0.96	0.0160579	0.0069237	0.280562
3	6	0.197351	0.256401	-0.78	0.0348365	0.00907587	0.623824
3	7	0.0139561	0.0176754	0.304	0.00311266	0.0026567	0.907125
3	8	0.193923	0.0679952	-0.236	0.352445	0.00476901	0.83852
3	9	0.304918	0.313804	0.976	0.149346	0.00913315	0.177624
3	10	0.155131	0.230993	0.728	0.0465985	0.00894827	0.681081
3	11	0.0200062	0.0275847	-0.352	0.00573225	0.00319739	0.783935
3	12	0.363974	0.395876	-0.584	0.0947579	0.0100034	0.813382
3	13	0.36742	0.465823	0.952	0.078047	0.00971212	0.306712
3	14	0.0380752	0.0480191	0.428	0.00997801	0.00443259	0.867077
3	15	0.337402	0.194535	1.	0.16222	0.0077413	0.
3	16	0.0189958	0.0258282	0.756	0.00358338	0.00303184	0.621295
3	17	0.0181767	0.0189547	-0.58	0.00381533	0.0028211	0.610813
3	18	0.515572	0.536659	0.68	0.136694	0.0103298	0.734683
3	19	0.00718842	0.00778998	1.	0.00179017	0.00186209	0.
3	20	0.115846	0.142926	0.624	0.0202774	0.00771804	0.777846
3	21	0.553188	0.623957	-0.456	0.115252	0.0089329	0.88725
3	22	0.254178	0.265862	1.	0.0130487	0.0095385	0.
3	23	0.606714	0.447518	0.992	0.224906	0.00954381	0.126491
3	24	0.0153853	0.0379475	0.592	0.00360782	0.00383523	0.75619
3	25	0.105646	0.240162	0.336	0.0625697	0.00907091	0.935202
3	26	0.647916	0.551069	-0.572	0.224783	0.00955271	0.819452
3	27	0.290034	0.387613	0.068	0.0677015	0.00904859	0.989592
3	28	0.544848	0.611399	0.08	0.114675	0.0098627	0.998794
3	29	0.465568	0.607031	0.076	0.0543937	0.00989866	0.99306
3	30	0.691543	0.62442	0.856	0.197871	0.0106483	0.518013
3	31	0.976512	0.729933	1.	0.0703032	0.0084034	0.
3	32	0.719277	0.655379	0.992	0.211215	0.0099021	0.126491
3	33	0.0414954	0.0602005	-0.02	0.0109354	0.00491716	0.9836

MatrixForm[ZZ[[4]]]

4	1	0.068173	0.117956	0.524	0.0180637	0.00423139	0.831981
4	2	0.365167	0.375245	1.	0.00921299	0.00688979	0.
4	3	0.181947	0.212293	-0.396	0.0693253	0.00533202	0.904687
4	4	0.0950354	0.0929644	1.	0.00444644	0.0041322	0.
4	5	0.348092	0.423619	1.	0.0375857	0.00692362	0.
4	6	0.232612	0.312615	-0.376	0.056242	0.00615298	0.924143
4	7	0.00359728	0.00443933	-0.484	0.000819057	0.000888413	0.500746
4	8	0.0521529	0.102946	-0.584	0.0130651	0.00407003	0.788308
4	9	0.186956	0.224918	0.092	0.0674076	0.00568683	0.995742
4	10	0.0789004	0.101337	1.	0.0115404	0.00380515	0.
4	11	0.0283012	0.0344582	-0.472	0.00838272	0.0022659	0.776937
4	12	0.141482	0.180711	0.404	0.0416357	0.00489625	0.909998
4	13	0.276214	0.38286	0.728	0.0474654	0.00593027	0.681081
4	14	0.0601063	0.0676946	0.904	0.00830446	0.00324806	0.42839
4	15	0.0606041	0.0324435	1.	0.0870641	0.00228955	0.
4	16	0.00972066	0.0107887	1.	0.00139921	0.00141338	0.
4	17	0.0119875	0.0125335	-0.736	0.00168631	0.00133925	0.485021
4	18	0.32108	0.352249	1.	0.011253	0.00660904	0.
4	19	0.0522474	0.0635084	0.532	0.0123929	0.00335288	0.831711
4	20	0.0980777	0.118774	0.776	0.0151744	0.00429701	0.625611
4	21	0.527415	0.588787	0.976	0.114924	0.00625455	0.218207
4	22	0.269292	0.261186	1.	0.013038	0.0059849	0.
4	23	0.139604	0.150946	0.688	0.0508206	0.00486984	0.721622
4	24	0.0516479	0.0599895	0.796	0.132312	0.00340834	0.575943
4	25	0.0299721	0.0544289	0.64	0.00535031	0.00293322	0.759412
4	26	0.254065	0.289297	0.256	0.120911	0.00660745	0.968616
4	27	0.163821	0.184205	1.	0.0154896	0.00514136	0.
4	28	0.482484	0.489697	0.992	0.136184	0.00622417	0.126491
4	29	0.774226	0.706172	0.92	0.12651	0.00591554	0.392705
4	30	0.583407	0.591192	0.42	0.169256	0.00726733	0.907134
4	31	0.864467	0.69622	0.976	0.13427	0.0065743	0.218207
4	32	0.169373	0.230412	-0.468	0.0231315	0.00578787	0.878672
4	33	0.0250596	0.0328431	-0.516	0.00451266	0.00245104	0.841771

MatrixForm[ZZ[[5]]]

5	1	0.050022	0.0739264	-0.116	0.0127777	0.00617643	0.964502
5	2	0.174371	0.257314	0.744	0.0259976	0.00965471	0.663494
5	3	0.096036	0.109803	-0.304	0.0434841	0.00664017	0.920311
5	4	0.443185	0.435482	0.992	0.0293292	0.0110768	0.126491
5	5	0.0697618	0.129555	0.288	0.0213609	0.00764283	0.938391
5	6	0.224159	0.335452	-0.068	0.0760453	0.00999829	0.997676
5	7	0.0174204	0.0223093	0.636	0.00405142	0.00331233	0.722212
5	8	0.0113065	0.0223623	0.592	0.00481775	0.00357297	0.492449
5	9	0.457912	0.413897	-0.568	0.202646	0.0111181	0.82468
5	10	0.11511	0.172071	0.208	0.0380088	0.00841556	0.96772
5	11	0.0903915	0.0868395	-0.684	0.082151	0.0064451	0.700075
5	12	0.234029	0.278999	0.44	0.0783068	0.0105261	0.890828
5	13	0.367802	0.486174	0.28	0.0453344	0.00998472	0.961926
5	14	0.0322612	0.0435464	0.892	0.00556747	0.00455642	0.448487
5	15	0.30476	0.183758	0.892	0.125513	0.00906234	0.311003
5	16	0.0257422	0.0324713	0.8	0.00454144	0.00371624	0.56682
5	17	0.00970086	0.00974374	0.056	0.00233318	0.00208516	0.230383
5	18	0.283002	0.30271	1.	0.0131397	0.00969849	0.
5	19	0.100737	0.111075	0.788	0.0193618	0.00704697	0.607067
5	20	0.12565	0.152937	0.804	0.023201	0.00778241	0.592442
5	21	0.594443	0.634545	-0.12	0.136701	0.00771716	0.994765
5	22	0.197698	0.229508	0.744	0.0420532	0.00964363	0.663494
5	23	0.266321	0.288748	0.736	0.0503392	0.010079	0.67834
5	24	0.0176789	0.0422739	0.788	0.00366396	0.00459521	0.586885
5	25	0.170201	0.244778	0.956	0.0236334	0.00982365	0.287043
5	26	0.594863	0.569225	0.456	0.211015	0.00825368	0.891765
5	27	0.190255	0.230351	0.824	0.0274775	0.0101274	0.560608
5	28	0.505866	0.557367	0.564	0.114551	0.0108257	0.825001
5	29	0.296573	0.426038	0.764	0.0436177	0.010622	0.63079
5	30	0.781002	0.660024	0.872	0.185667	0.0102304	0.490488
5	31	0.868895	0.679753	0.936	0.152306	0.0102256	0.34113
5	32	0.169625	0.234645	-0.312	0.0254658	0.00926235	0.94776
5	33	0.029727	0.0416789	-0.216	0.006408	0.00474481	0.961792

MatrixForm[ZZ[[6]]]

6	1	0.0533843	0.083913	-0.612	0.00971437	0.00517363	0.764062
6	2	0.416891	0.445072	1.	0.0337646	0.00933191	0.
6	3	0.0830026	0.087636	-0.58	0.0326588	0.00527655	0.788726
6	4	0.376175	0.351691	1.	0.0202061	0.00876701	0.
6	5	0.0661148	0.103209	-0.16	0.0164579	0.0053064	0.968583
6	6	0.193423	0.253769	-0.76	0.0404454	0.00866059	0.64503
6	7	0.0125501	0.0156954	0.624	0.00271992	0.00236759	0.735382
6	8	0.0119497	0.0228447	0.456	0.00428602	0.00264437	0.499059
6	9	0.385355	0.404257	-0.688	0.0993007	0.00849859	0.727166
6	10	0.114263	0.152368	-0.436	0.0281556	0.00681445	0.899522
6	11	0.0563618	0.0608269	-0.908	0.0138981	0.00481742	0.395172
6	12	0.201669	0.244657	-0.488	0.0337766	0.00742964	0.869991
6	13	0.397177	0.516206	0.268	0.0689801	0.00843024	0.963269
6	14	0.0203717	0.0311533	0.904	0.00344486	0.00323318	0.409211
6	15	0.283782	0.158849	0.916	0.153286	0.00710952	0.277944
6	16	0.0184124	0.0241978	0.684	0.00334332	0.00273816	0.694314
6	17	0.0126423	0.0132344	-0.48	0.00253722	0.00207792	0.57526
6	18	0.481544	0.517816	1.	0.0161464	0.00985228	0.
6	19	0.13079	0.140708	0.788	0.0223155	0.00663066	0.600415
6	20	0.0588987	0.0735786	0.736	0.0102275	0.00484623	0.666394
6	21	0.498733	0.582006	0.768	0.117737	0.00930591	0.641735
6	22	0.236616	0.269887	0.752	0.0470423	0.00770536	0.654377
6	23	0.31943	0.366449	-0.076	0.116889	0.00921554	0.99306
6	24	0.0346035	0.0491078	0.484	0.10279	0.00373425	0.827334
6	25	0.109834	0.158647	0.612	0.0187138	0.00688702	0.779671
6	26	0.644146	0.608071	-0.416	0.174723	0.00886018	0.911189
6	27	0.155228	0.208103	0.504	0.0329596	0.00815205	0.856105
6	28	0.598918	0.561919	0.956	0.172533	0.00943091	0.287043
6	29	0.391617	0.475137	1.	0.0526091	0.0102949	0.
6	30	0.621084	0.664732	-0.116	0.162485	0.00811799	0.993222
6	31	0.863922	0.667418	1.	0.135984	0.00862969	0.
6	32	0.241368	0.329848	-0.136	0.0351212	0.00854383	0.992696
6	33	0.0653021	0.0773492	-0.956	0.0113775	0.00463091	0.287043

MatrixForm[ZZ[[7]]]

7	1	0.0988958	0.113758	1.	0.00738028	0.00613335	0.
7	2	0.348513	0.450999	0.976	0.0512568	0.0102063	0.218207
7	3	0.390186	0.455937	-0.132	0.151346	0.011407	0.983078
7	4	0.228131	0.219653	1.	0.0110898	0.0076886	0.
7	5	0.31939	0.492518	0.132	0.0689099	0.0106642	0.991214
7	6	0.852964	0.697201	-0.916	0.148065	0.00855715	0.396956
7	7	0.00860398	0.0111941	0.7	0.00208581	0.00207185	0.641597
7	8	0.492218	0.368928	-0.8	0.245401	0.0104761	0.601204
7	9	0.532854	0.541387	0.56	0.200165	0.010641	0.830155
7	10	0.398755	0.441968	-0.948	0.0289776	0.0107605	0.312549
7	11	0.0439774	0.0471404	-0.768	0.00893921	0.00424351	0.582696
7	12	0.391077	0.434286	0.544	0.164595	0.0106113	0.840768
7	13	0.686703	0.719018	-0.476	0.156972	0.00880856	0.878928
7	14	0.465107	0.471663	-1.	0.0126542	0.0097722	0.
7	15	0.148406	0.0758388	1.	0.135578	0.00576808	0.
7	16	0.0257351	0.0451819	-0.044	0.00707676	0.00425859	0.957985
7	17	0.0905266	0.129729	-0.724	0.0235312	0.00699173	0.664523
7	18	0.924951	0.588376	1.	0.110302	0.0105768	0.
7	19	0.456563	0.463404	0.904	0.0604775	0.0108225	0.42839
7	20	0.756715	0.638891	0.872	0.177543	0.00934932	0.490488
7	21	0.767665	0.744391	0.608	0.182904	0.0082756	0.79553
7	22	0.650079	0.467697	1.	0.144913	0.0104585	0.
7	23	0.508246	0.421089	0.996	0.220054	0.00987891	0.0632456
7	24	0.0708973	0.0986813	0.896	0.14609	0.00626564	0.426511
7	25	0.202851	0.357431	0.668	0.0959091	0.00988456	0.742956
7	26	0.694723	0.514012	-0.048	0.300836	0.00992243	1.00085
7	27	0.373633	0.437822	0.78	0.0676734	0.0111524	0.623824
7	28	0.815277	0.717241	0.252	0.190615	0.0081958	0.967595
7	29	0.587725	0.69157	0.364	0.078576	0.00988554	0.92679
7	30	0.905808	0.768488	0.712	0.139241	0.00715382	0.697857
7	31	0.881307	0.760601	0.908	0.132088	0.00850165	0.415
7	32	0.436926	0.54464	0.888	0.0917363	0.010135	0.460766
7	33	0.0523231	0.0806398	0.664	0.0193699	0.00545748	0.732975

MatrixForm[ZZ[[8]]]

8	1	0.0716637	0.107881	-0.42	0.0145986	0.00697262	0.893753
8	2	0.187357	0.278125	-0.184	0.0240222	0.0100864	0.980812
8	3	0.189871	0.1457	-0.772	0.169721	0.00790211	0.620933
8	4	0.413965	0.342119	1.	0.0421979	0.0102849	0.
8	5	0.100831	0.174419	0.404	0.0277923	0.00873978	0.905574
8	6	0.248484	0.320144	-0.7	0.0474998	0.0110237	0.707107
8	7	0.00674465	0.0120688	0.848	0.00200841	0.00257823	0.457968
8	8	0.0158102	0.0347188	0.812	0.00630471	0.00419975	0.391496
8	9	0.416478	0.398019	-0.536	0.16197	0.0116115	0.845911
8	10	0.224418	0.272056	-0.78	0.0303376	0.0103036	0.623824
8	11	0.0459218	0.0546687	-0.768	0.00997206	0.0051803	0.622677
8	12	0.169799	0.209319	0.384	0.0465726	0.00966172	0.925185
8	13	0.367581	0.489969	0.156	0.0556885	0.0111144	0.983633
8	14	0.0462796	0.0532813	0.924	0.00779793	0.00547256	0.377882
8	15	0.385482	0.220263	1.	0.171772	0.00954176	0.
8	16	0.0170466	0.02065	0.996	0.0037692	0.00332613	0.0632456
8	17	0.0234623	0.0245313	-0.736	0.00532258	0.0037162	0.575874
8	18	0.317437	0.345963	1.	0.0161046	0.0109848	0.
8	19	0.23231	0.283688	0.08	0.0699367	0.0106166	0.994765
8	20	0.234502	0.305256	-0.112	0.0593026	0.01055	0.99166
8	21	0.429553	0.531456	0.8	0.106939	0.0116946	0.601204
8	22	0.255362	0.307244	0.516	0.0672516	0.0110209	0.855964
8	23	0.286643	0.343819	-0.052	0.065102	0.0115248	0.998642
8	24	0.028928	0.046125	0.972	0.00512854	0.00498098	0.226764
8	25	0.0699556	0.1192	0.156	0.0165585	0.00693508	0.963002
8	26	0.556951	0.513494	-0.688	0.179392	0.0109924	0.727166
8	27	0.164324	0.2106	0.78	0.0236109	0.00816915	0.617353
8	28	0.578957	0.60205	0.312	0.164374	0.0113141	0.951988
8	29	0.380883	0.487931	0.88	0.0512504	0.011573	0.467412
8	30	0.724017	0.66155	0.504	0.183312	0.0113141	0.865436
8	31	0.811566	0.677481	0.904	0.167276	0.0104147	0.41891
8	32	0.17877	0.245269	-0.4	0.0258681	0.00983179	0.91397
8	33	0.0332446	0.0482875	-0.284	0.0080712	0.00511433	0.937475

We then compute expansion times (generations) with SD for each marker

Out[8]//MatrixForm=

1	473.619	39.963	737.705	55.2901	66.4091	22.2708	136.371	45.0339	103.906	16.2595	94.8172	25.9857	147.632	11.8968	137.647	33.3755
2	125.825	33.9499	172.77	22.3832	95.2582	24.3623	206.561	7.69097	91.5539	23.9092	253.233	33.103	194.949	42.8185	130.839	24.2592
3	95.1727	170.344	111.213	218.702	619.322	347.207	172.453	44.9613	85.8781	21.9278	68.2596	21.2389	462.05	86.72	121.301	137.652
4	88.307	9.3999	93.6473	23.6998	142.394	79.0635	19.6585	0.990405	116.795	10.7682	88.1661	6.42315	50.1844	2.89633	85.1478	13.2225
5	140.995	28.7181	141.591	30.9967	114.693	16.8625	374.482	56.6992	96.4498	28.2143	82.5169	14.5067	591.139	138.817	127.59	40.3863
6	172.164	17.2196	169.968	42.5221	78.376	17.2838	111.98	25.9626	135.531	19.3964	77.3145	19.1101	482.026	303.226	105.817	25.2597
7	2875.86	490.739	2869.09	507.439	258.718	51.9884	62.4645	14.8155	300.835	71.0205	214.295	47.4112	145.916	37.5315	117.338	44.4137
8	166.571	30.4803	198.83	41.6867	328.999	919.596	355.75	137.516	74.1543	32.9458	83.9109	30.6052	2225.74	1907.96	90.9775	45.1258
9	153.758	35.9503	132.329	39.6047	104.831	68.8247	70.1694	9.07422	160.775	96.1034	147.847	46.6232	248.134	143.249	151.629	69.3032
10	112.347	16.7275	103.927	16.9701	103.352	35.3065	43.0487	6.66001	93.0332	19.5772	77.3145	17.0126	292.849	31.8968	145.244	25.138
11	750.745	188.169	654.011	160.961	69.7364	15.8727	88.633	19.246	251.794	207.854	161.77	39.3838	126.531	23.8092	135.949	28.5695
12	172.623	52.4247	153.7	39.7006	179.753	52.124	63.3483	14.4725	108.643	30.1672	90.6071	15.8658	204.059	97.2977	76.5448	16.4531
13	116.11	45.7151	136.94	65.1527	67.0911	21.6276	50.6446	14.1472	91.8649	20.8931	103.465	25.7235	270.388	239.089	98.3154	19.1179
14	180.157	37.1396	150.361	34.4954	113.017	23.4628	158.862	22.4543	85.8909	17.015	54.4972	12.053	1709.88	70.7437	121.129	20.9777
15	97.3955	75.2826	93.3245	74.5596	72.6501	68.6249	11.0698	30.2874	68.1842	51.6098	58.0841	60.8758	26.4749	49.619	83.56	75.2877
16	149.185	31.7161	130.906	40.0869	128.427	28.6008	59.9488	8.68852	169.141	32.7761	125.765	26.1803	280.274	38.8735	105.768	23.6839
17	89.301	26.9175	114.724	34.7964	163.405	30.8137	106.765	14.3504	85.618	18.5531	114.019	20.6466	937.913	263.46	210.289	44.6044
18	94.9748	7.57	96.6429	9.38199	217.143	81.2287	107.506	4.87222	91.6038	5.29606	190.39	10.0279	267.88	98.0394	105.934	6.92489
19	138.922	23.583	149.495	26.5875	7.25278	1.8138	59.8099	11.7217	110.123	20.5258	144.982	24.2743	647.997	119.724	338.822	38.7737
20	92.1457	15.6538	113.315	20.2347	118.502	21.9077	94.8901	15.958	123.648	24.649	56.3249	10.4852	1100.42	796.985	323.756	44.1141
21	142.072	32.3578	140.213	53.0375	107.728	41.5429	83.7806	40.3328	127.013	29.3954	80.3458	36.2971	238.826	487.68	61.9723	25.6574
22	154.002	12.9221	130.188	12.5952	114.658	7.04987	118.465	7.12526	89.5252	19.6192	109.666	23.1405	255.749	122.722	129.174	31.3239
23	165.236	141.401	148.968	106.385	165.996	139.899	40.1872	13.2411	84.8021	17.2312	122.985	16.4799	151.642	126.41	113.056	9.92815
24	191.894	40.5842	192.922	60.73	77.9685	37.6822	172.509	399.444	72.5033	34.6534	135.041	238.175	242.036	484.27	93.5511	19.0032
25	85.7219	22.6325	73.0807	21.3695	159.999	70.2137	28.3434	9.46746	139.944	25.5022	100.367	24.1639	219.248	114.327	85.8124	16.7109
26	113.641	105.139	122.407	91.731	125.814	99.2708	44.0407	12.9445	134.593	82.4103	156.076	77.3584	109.856	31.5922	109.649	66.6078
27	402.666	259.62	355.639	223.569	247.521	29.3767	86.3376	9.26758	106.963	18.7561	97.0463	22.0823	253.885	62.6625	92.7872	17.4268
28	153.753	54.4328	137.977	22.3167	122.064	19.4109	75.9209	41.6405	90.1053	32.0355	102.431	83.6563	205.107	152.274	116.625	47.1081
29	114.045	20.0969	107.852	21.623	277.724	48.8685	469.822	491.633	96.59	27.8727	125.613	26.1451	358.205	127.404	126.658	29.8169
30	122.051	115.054	145.161	132.568	94.4112	109.03	81.4643	38.9039	110.092	142.554	109.917	29.6707	197.118	303.208	110.851	86.9659
31	123.396	84.7769	126.155	112.625	143.976	47.6399	124.774	84.1168	116.769	90.7519	111.253	69.3788	165.815	138.743	115.724	97.8788
32	86.5257	18.4559	88.5065	20.7941	557.537	1149.59	82.3728	16.87	88.5457	17.1071	146.552	21.461	246.44	88.7909	90.7087	18.8069
33	162.454	28.7201	191.99	40.0415	144.285	14.4656	67.9482	12.8575	92.8509	17.3979	159.49	29.0914	149.361	52.9242	104.91	22.4546

Times tj for each marker (and error spread) is graphed for each SNP

(8) P312

The individual branches may be combined to obtain the spectrum of branching times

(7) J2 : Branching Spectrum

We compute the mean time, ie average generations over branches *30 ybp:

Out[14]//MatrixForm=

$$\begin{pmatrix} 7673.63 \\ 8085.01 \\ 4628.99 \\ 3527.7 \\ 3506.23 \\ 3651.16 \\ 12485.5 \\ 3867.6 \end{pmatrix}$$

Next we go through the stochastic simulation which generates the following

TMRCAs

Out[11]//MatrixForm=

SNP	TMRCAs ybp	SD
G2a2*	7654	1007
R1b1a2	6004	701
R1a1a	6016	741
I1	4109	460
L21	3927	415
U106	4883	534
J2	14 490	2000
P312	4449	490

* Western Europe

SI 7: data for R1b1a2 for DYS j= 1, .. 33

Use FTDNA R1b1a2 data, which has been corrected (for missing files etc) and counted

12	24	14	10	11	13	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	11	9	19	23	17	16	18	17	37	38	13	12	
12	24	14	10	11	13	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	12	10	19	23	17	16	17	18	38	39	12	12	
12	24	14	10	11	14	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	11	10	19	23	17	16	17	17	38	39	12	12	
12	24	14	10	11	14	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	11	10	19	23	17	16	17	17	38	40	12	12	
12	24	14	10	11	14	11	12	12	14	14	16	17	9	10	11	11	24	15	19	29	11	10	19	23	17	16	18	17	37	38	12	12	
12	24	14	11	11	14	11	12	12	13	13	16	15	9	10	11	11	25	15	20	28	10	11	19	22	16	14	18	17	37	37	12	12	
12	24	14	9	11	14	11	12	12	13	14	16	17	9	9	11	11	25	15	19	30	11	10	19	23	17	16	17	17	38	38	13	12	
12	24	15	10	11	14	11	13	13	13	13	16	19	9	10	11	11	24	15	19	27	11	11	19	23	15	15	17	17	35	37	13	12	
12	25	14	10	11	14	11	12	11	13	13	16	14	10	10	11	11	25	15	19	30	10	11	19	22	15	16	20	17	37	38	11	12	
12	25	14	10	12	15	11	12	13	13	13	16	16	9	10	11	11	25	15	20	33	10	10	19	23	16	15	18	17	35	38	12	12	
12	25	14	10	14	14	11	12	11	13	13	16	17	9	9	11	11	24	15	19	31	11	11	17	23	15	17	19	18	35	38	13	12	
12	25	14	11	11	11	11	12	12	13	13	16	17	9	10	11	11	25	15	19	28	10	11	18	23	14	15	21	18	34	37	12	12	
12	25	14	11	11	14	11	12	12	13	13	16	17	9	10	11	11	25	15	19	28	10	11	18	23	14	15	21	18	34	37	12	12	
12	25	14	11	11	15	11	12	12	14	14	16	17	9	10	11	11	25	15	19	28	10	11	18	23	14	15	20	18	34	38	12	12	
12	25	14	11	11	15	11	12	12	12	14	16	16	9	9	11	11	25	16	19	32	11	11	19	23	17	14	18	18	35	38	12	12	
12	25	14	11	12	14	11	12	11	14	13	16	16	9	10	11	11	25	15	19	28	11	11	19	23	15	15	20	18	35	37	13	12	
12	25	14	10	11	15	11	12	13	13	13	16	19	9	10	11	11	25	15	19	29	10	10	19	22	15	17	17	16	35	39	12	12	
12	24	14	10	11	13	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	11	9	19	23	17	16	18	17	37	38	12	12	
12	24	14	10	11	14	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	11	10	19	23	17	16	17	17	38	39	12	12	
12	24	14	10	11	14	11	12	12	13	14	16	17	9	10	11	11	25	15	19	30	12	10	19	23	16	16	17	18	38	39	12	12	
12	24	14	10	11	14	11	12	12	14	14	16	17	9	9	11	11	24	15	19	29	11	10	19	23	17	16	17	17	38	38	12	12	
12	24	14	10	11	14	11	12	12	13	13	16	19	9	9	11	11	25	15	19	31	11	9	11	19	21	15	16	19	17	36	37	12	12
12	24	14	11	11	14	11	12	12	13	13	16	16	9	10	11	11	25	15	19	31	10	12	19	23	15	16	20	18	35	37	12	12	
12	25	14	10	11	14	11	12	13	13	13	17	16	9	10	11	11	24	16	19	28	10	11	19	23	16	16	21	18	36	39	12	12	
12	25	14	11	10	15	11	12	11	13	13	16	19	9	9	11	11	24	15	19	28	10	10	17	23	16	16	18	17	35	37	12	12	
12	25	14	11	10	15	11	12	12	14	13	16	16	9	10	11	11	24	15	19	28	10	10	17	23	16	16	18	17	35	37	12	12	
12	24	14	10	10	14	12	12	13	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	15	17	18	17	36	38	12	12	
12	24	14	10	11	14	12	12	12	13	13	16	19	9	10	11	11	25	15	19	29	10	11	18	23	15	17	21	16	36	38	12	12	
12	24	14	10	11	14	12	12	12	13	13	16	20	9	10	11	11	25	15	19	29	10	11	18	23	15	17	21	16	36	38	12	12	
12	24	14	10	11	14	12	12	12	13	13	16	17	9	10	11	11	25	15	19	30	10	11	18	23	15	17	20	16	36	38	12	12	
12	23	14	11	11	14	12	12	11	14	13	16	17	9	9	11	11	26	15	19	28	11	12	19	23	15	15	17	16	36	38	12	12	
12	23	14	11	11	14	12	12	12	13	13	17	17	9	10	11	11	25	15	19	28	11	12	19	23	15	15	19	18	36	37	12	12	
12	23	14	11	11	14	12	12	12	13	13	17	18	9	10	11	11	26	15	19	28	11	12	19	23	15	15	17	17	37	37	11	12	
12	24	13	10	11	14	12	12	12	14	13	16	17	9	10	11	11	26	16	19	30	11	11	19	22	16	16	16	16	35	39	12	12	
12	24	14	10	10	14	12	11	12	13	13	17	16	9	10	11	11	25	15	19	29	10	11	19	22	15	16	18	17	37	37	12	12	
12	24	14	10	10	14	12	11	12	13	13	17	16	9	10	11	11	24	15	19	29	10	12	19	23	15	15	18	16	36	37	12	12	
12	24	14	10	11	14	12	11	12	13	13	17	16	9	10	11	11	24	15	19	29	10	12	19	23	15	15	18	16	36	39	12	12	
12	24	14	10	11	14	12	11	12	13	13	17	19	9	9	11	11	24	15	19	29	10	12	19	23	15	15	18	16	37	39	12	12	
12	24	14	10	12	15	12	12	12	14	14	15	9	9	10	11	11	25	15	19	29	12	10	19	23	15	15	18	18	35	35	12	12	
12	24	14	11	11	13	12	12	13	13	13	16	16	9	10	11	11	25	15	20	28	12	11	19	19	16	15	17	17	36	38	12	12	
12	24	14	11	11	13	12	12	13	13	13	16	16	9	10	11	11	25	15	20	28	12	11	19	19	16	15	17	17	37	38	12	12	
12	24	14	11	11	14	12	12	12	13	13	16	15	9	10	11	11	25	15	19	28	11	11	19	23	16	16	17	16	36	37	11	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	14	19	28	11	11	19	23	16	16	17	16	36	37	11	12	
12	24	14	11	11	14	12	12	12	13	13	17	17	9	10	11	11	25	14	19	28	11	11	19	23	16	16	17	16	36	37	11	12	
12	24	14	11	11	14	12	12	12	13	13	17	15	9	10	11	11	25	15	19	29	12	11	19	23	15	15	15	17	37	39	12	12	
12	24	14	10	11	14	12	12	12	13	13	16	16	9	9	11	11	25	15	19	27	11	12	19	21	15	15	17	17	36	36	12	10	
12	24	14	11	11	14																												

12	24	14	11	11	14	12	13	12	13	13	16	17	9	9	11	11	24	16	19	29	11	11	19	23	15	17	19	18	38	41	12	12	
12	24	14	11	11	14	12	12	12	13	13	16	17	9	9	11	11	24	16	19	30	11	11	19	23	15	17	18	18	37	37	43	12	12
12	24	14	11	11	14	12	12	12	13	13	16	17	9	10	11	11	26	15	18	29	12	12	19	23	14	16	18	17	37	38	13	12	
12	24	15	10	11	14	12	12	12	13	12	16	16	9	10	11	11	24	15	19	29	10	11	19	23	16	15	19	17	36	37	12	12	
12	24	14	10	11	14	12	13	12	13	13	16	14	8	10	11	11	24	15	18	27	11	12	19	23	15	16	19	17	38	40	13	12	
12	24	14	11	11	15	12	12	12	13	13	16	17	9	9	11	11	24	16	19	30	10	10	19	23	15	17	18	18	37	41	13	12	
12	24	14	11	11	15	12	12	12	13	13	16	17	9	9	11	11	24	16	19	29	10	11	19	23	15	17	19	18	36	40	13	12	
12	24	14	11	11	15	12	12	12	13	13	16	17	9	9	11	11	24	16	19	30	11	11	19	23	15	17	19	18	37	42	13	12	
12	23	14	11	11	14	12	12	12	14	13	17	20	9	9	11	11	26	15	19	31	11	12	19	23	15	17	20	17	37	38	12	12	
12	24	14	10	11	14	12	12	12	13	13	17	16	9	10	11	11	25	14	19	30	11	11	19	23	16	16	18	18	37	40	13	12	
12	24	14	11	11	13	12	12	12	13	13	16	16	9	10	11	11	25	14	19	30	10	10	19	23	16	16	19	18	36	39	12	12	
12	24	14	11	11	14	11	12	13	13	13	16	16	9	9	11	11	23	15	19	29	11	11	19	23	15	17	19	18	37	38	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	16	16	18	17	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	16	16	18	18	36	38	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	16	16	18	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	11	19	23	15	16	18	18	37	38	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	15	16	18	18	37	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	16	16	18	17	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	9	11	11	25	15	19	30	11	12	19	23	16	16	18	18	37	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19	23	15	16	19	18	36	37	12	12	
 	
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	30	11	12	19										

12	24	14	10	11	14	12	12	13	13	13	16	17	9	10	11	11	26	15	19	30	10	12	18	23	15	17	19	16	35	39	12	12	
12	24	14	10	11	14	12	12	13	13	13	16	17	9	9	11	11	26	15	19	30	10	12	18	23	15	17	19	16	35	39	13	12	
12	24	14	10	11	15	12	12	12	13	13	16	17	9	9	11	11	24	16	19	31	11	11	19	21	15	15	17	17	36	36	12	12	
12	24	14	11	11	16	13	12	12	13	13	17	19	9	9	11	11	25	15	20	30	10	11	19	23	15	14	17	17	37	37	11	12	
13	23	14	11	11	15	13	12	12	13	13	17	16	9	10	11	11	25	15	18	32	11	11	19	23	15	15	17	16	38	38	13	12	
13	23	14	11	11	12	14	13	12	12	14	16	17	9	10	10	11	25	15	19	31	11	11	19	22	15	15	18	17	36	37	12	12	
13	23	14	11	11	13	14	13	12	12	13	16	17	9	10	10	11	27	15	19	29	11	9	19	23	15	15	19	17	34	39	12	12	
13	23	14	11	11	13	14	13	12	12	13	16	17	9	10	10	11	27	15	19	29	11	9	19	23	15	15	19	17	34	39	12	12	
13	24	13	10	11	14	13	12	12	13	14	16	17	9	10	10	11	25	17	19	30	10	12	19	23	17	14	17	17	38	38	10	12	
13	24	14	11	11	14	13	12	11	13	13	15	17	9	10	11	11	25	15	19	30	11	11	19	23	16	15	18	17	35	37	12	12	
13	24	14	11	11	14	13	12	11	13	13	16	18	9	10	11	11	25	15	19	29	11	11	19	23	16	15	20	17	37	38	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	28	11	11	19	23	16	15	17	18	37	38	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	15	9	10	11	11	24	16	19	29	11	11	19	23	16	15	19	19	35	38	11	12	
13	25	14	10	11	14	13	12	12	12	13	15	17	9	10	11	10	24	15	19	29	11	11	19	23	16	15	18	17	36	39	12	12	
13	25	14	11	11	14	13	12	12	13	13	16	16	9	10	11	11	24	15	19	30	11	11	19	23	15	17	17	17	36	38	12	12	
13	25	14	11	11	14	13	12	13	13	13	16	16	9	10	11	11	24	15	19	30	11	11	19	23	15	17	17	17	36	37	12	12	
14	23	14	11	11	12	15	13	12	12	13	13	17	17	9	10	11	11	24	15	19	31	12	11	19	23	16	16	18	38	38	12	12	
13	23	14	11	11	14	13	12	12	13	13	15	20	9	10	11	11	24	15	19	30	10	12	19	23	15	15	18	19	36	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	14	15	17	9	10	11	11	23	15	19	30	11	11	19	23	18	15	20	16	33	37	13	12
13	24	14	11	11	12	14	13	12	13	13	14	15	17	9	11	11	23	15	19	30	11	11	19	23	18	15	19	17	34	36	13	12	
13	24	14	11	11	12	15	13	12	12	14	13	15	18	9	10	11	11	25	15	19	28	11	11	19	23	16	15	16	16	36	36	12	12
13	24	14	11	11	14	13	12	13	13	14	15	17	9	10	11	11	23	15	19	30	11	9	19	23	17	15	17	17	34	36	13	11	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	17	36	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	18	36	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	18	36	37	12	12	
13	24	14	10	11	16	13	12	12	13	13	16	16	9	10	11	11	25	15	19	29	11	11	21	23	16	15	18	16	36	36	12	13	
13	24	14	11	11	14	13	12	11	13	13	16	16	9	11	10	11	23	16	19	30	11	10	19	23	15	14	16	16	38	38	11	12	
13	24	14	11	11	14	13	12	12	13	13	16	16	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	18	38	40	12	12	
13	24	14	11	11	15	13	12	12	13	13	16	16	9	10	11	11	26	15	20	29	11	11	19	23	16	15	18	17	38	39	12	12	
13	24	14	11	11	15	13	12	12	13	13	16	16	9	10	11	11	26	15	20	29	11	11	19	23	16	15	18	17	38	39	12	12	
13	24	14	11	11	12	14	13	12	12	13	13	17	10	9	10	11	11	25	15	19	28	11	11	19	23	16	15	16	18	37	38	12	12
13	24	14	11	11	14	13	12	12	13	14	15	18	9	10	11	11	25	15	19	30	11	11	19	23	15	14	18	14	18	37	37	12	11
13	24	14	10	11	14	13	12	12	13	13	15	17	9	10	11	11	24	15	19	30	11	11	19	23	16	15	18	17	38	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	15	17	9	10	11	11	25	15	19	28	11	11	19	23	16	15	18	17	38	37	12	12	
13	24	14	11	11	14	13	12	13	13	14	15	17	9	11	11	11	23	15	19	30	11	11	19	23	18	15	19	18	34	36	12	12	
13	24	15	11	11	13	13	12	12	13	13	15	19	9	10	11	11	25	15	19	28	10	10	19	23	16	16	19	19	34	37	12	11	
14	24	14	11	11	14	13	12	12	13	13	15	19	9	10	11	11	26	15	20	29	11	11	19	23	16	14	18	17	36	38	12	12	
14	24	14	11	11	14	13	12	13	13	13	15	18	9	10	11	12	25	14	19	31	10	11	19	23	16	15	18	17	35	36	12	12	
13	22	14	11	11	12	14	13	12	13	13	17	16	9	10	11	11	25	15	19	31	11	11	19	23	16	15	20	16	37	39	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	24	15	19	29	11	10	19	23	16	15	16	19	38	38	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	18	37	38	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	18	37	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	37	37	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19	29	11	11	19	23	16	15	17	19	36	36	12	12	
13	24	14	11	11	14	13	12	12	13	13	16	17	9	10	11	11	25	15	19														

13.	24.	14.	10.	11.	14.	12.	12.	12.	14.	14.	16.	17.	8.	10.	11.	11.	25.	15.	19.	30.	9.	11.	19.	23.	15.	16.	18.	18.	37.	40.	11.	12.	
13.	24.	14.	10.	11.	15.	12.	12.	10.	13.	13.	16.	17.	9.	10.	11.	11.	22.	15.	18.	30.	11.	11.	19.	23.	16.	15.	17.	16.	37.	42.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	11.	13.	13.	16.	16.	9.	10.	11.	11.	24.	15.	19.	30.	11.	11.	19.	23.	15.	15.	16.	18.	37.	37.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	11.	13.	13.	16.	17.	8.	10.	11.	11.	25.	15.	19.	31.	9.	11.	19.	22.	16.	15.	19.	18.	37.	38.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	11.	14.	13.	16.	16.	9.	10.	11.	11.	24.	15.	19.	30.	11.	11.	19.	23.	15.	15.	16.	18.	37.	37.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	12.	13.	16.	16.	9.	10.	11.	11.	25.	15.	19.	30.	11.	10.	19.	23.	14.	15.	17.	16.	37.	37.	12.	12.		
13.	24.	14.	11.	11.	14.	12.	12.	12.	13.	13.	15.	17.	9.	10.	11.	11.	25.	15.	19.	29.	11.	11.	19.	23.	16.	15.	19.	18.	35.	39.	10.	13.	
13.	24.	14.	11.	11.	14.	12.	12.	12.	13.	16.	16.	9.	10.	11.	11.	25.	15.	19.	30.	11.	11.	19.	23.	17.	15.	17.	18.	36.	36.	12.	12.		
13.	24.	14.	11.	11.	14.	12.	12.	13.	13.	12.	16.	18.	9.	10.	11.	11.	24.	15.	19.	30.	11.	10.	19.	22.	16.	15.	19.	17.	36.	40.	12.	12.	
13.	24.	14.	11.	11.	15.	12.	12.	12.	11.	13.	13.	16.	17.	9.	10.	11.	11.	26.	15.	19.	30.	11.	10.	19.	23.	16.	15.	17.	17.	36.	37.	12.	12.
13.	24.	14.	11.	11.	12.	14.	12.	12.	11.	13.	13.	18.	17.	9.	10.	10.	11.	25.	15.	19.	30.	11.	12.	19.	23.	16.	15.	16.	17.	36.	39.	12.	13.
13.	24.	14.	11.	11.	12.	14.	12.	12.	12.	13.	13.	16.	17.	9.	10.	10.	11.	25.	15.	19.	29.	11.	11.	19.	23.	17.	15.	18.	18.	36.	37.	12.	12.
13.	24.	14.	11.	11.	12.	15.	12.	12.	10.	13.	13.	16.	17.	8.	10.	11.	11.	26.	15.	19.	30.	11.	11.	19.	23.	16.	15.	18.	17.	35.	36.	12.	12.
13.	24.	14.	11.	11.	12.	15.	12.	12.	11.	13.	13.	16.	17.	9.	9.	11.	11.	25.	15.	19.	30.	11.	11.	19.	23.	17.	15.	16.	17.	37.	40.	12.	12.
13.	24.	14.	11.	11.	12.	15.	12.	12.	13.	13.	13.	17.	16.	9.	10.	11.	11.	26.	15.	19.	29.	11.	11.	18.	23.	15.	15.	17.	16.	37.	37.	12.	12.
13.	24.	14.	12.	11.	14.	12.	12.	12.	12.	12.	13.	16.	17.	9.	10.	11.	11.	25.	15.	19.	29.	11.	11.	19.	23.	16.	15.	18.	16.	38.	39.	12.	12.
13.	24.	15.	10.	11.	13.	12.	12.	12.	13.	13.	17.	18.	9.	9.	11.	11.	25.	15.	19.	31.	10.	11.	19.	23.	16.	15.	17.	17.	37.	37.	12.	12.	
13.	25.	14.	11.	11.	14.	12.	12.	12.	13.	13.	16.	16.	9.	10.	11.	11.	24.	15.	19.	29.	11.	11.	19.	20.	16.	15.	16.	17.	37.	38.	12.	12.	
13.	25.	14.	11.	11.	15.	12.	12.	12.	13.	13.	16.	16.	9.	10.	11.	11.	25.	15.	20.	29.	10.	11.	19.	23.	16.	15.	18.	16.	34.	37.	12.	12.	
13.	24.	14.	11.	11.	12.	12.	12.	12.	12.	14.	17.	16.	9.	10.	11.	11.	25.	15.	19.	30.	11.	11.	19.	23.	16.	15.	17.	16.	38.	41.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	12.	12.	14.	16.	17.	9.	10.	11.	11.	25.	15.	19.	29.	11.	11.	19.	23.	17.	15.	17.	17.	37.	39.	12.	12.	
13.	24.	14.	11.	11.	14.	12.	12.	12.	12.	14.	17.	16.	9.	10.	11.	11.	25.	15.	19.	30.	11.	11.	19.	23.	15.	15.	17.	17.	37.	38.	12.	12.	
13.	24.	14.	11.	11.	12.	14.	12.	12.	12.	14.	16.	16.	9.	10.	11.	11.	26.	14.	19.	31.	11.	11.	19.	23.	16.	15.	20.	17.	38.	39.	12.	12.	
13.	24.	15.	11.	11.	14.	12.	12.	12.	13.	14.	17.	17.	9.	10.	11.	11.	25.	15.	19.	31.	11.	11.	19.	23.	15.	15.	19.	17.	36.	37.	13.	13.	
12.	24.	14.	10.	11.	13.	11.	12.	12.	13.	14.	16.	16.	9.	10.	11.	11.	25.	15.	19.	29.	11.	9.	19.	23.	17.	16.	18.	17.	37.	38.	12.	12.	
12.	24.	14.	10.	11.	14.	11.	12.	12.	13.	14.	16.	17.	9.	10.	11.	11.	24.	15.	19.	29.	11.	10.	19.	23.	17.	16.	17.	17.	37.	38.	12.	12.	
12.	24.	14.	10.	11.	14.	11.	12.	12.	13.	14.	16.	17.	9.	10.	11.	11.	25.	15.	19.	30.	11.	10.	19.	23.	17.	16.	17.	17.	38.	38.	12.	12.	
12.	24.	14.	10.	11.	14.	11.	12.	12.	14.	14.	16.	17.	9.	10.	11.	11.	24.	15.	19.	29.	11.	10.	19.	23.	17.	16.	17.	17.	38.	38.	12.	12.	
12.	24.	14.	10.	11.	14.	11.	12.	12.	14.	14.	16.	16.	9.	10.	11.	11.	25.	15.	19.	29.	11.	10.	19.	23.	17.	16.	16.	18.	38.	40.	12.	12.	
12.	24.	14.	11.	11.	14.	11.	12.	11.	13.	13.	16.	16.	9.	10.	11.	11.	26.	15.	19.	28.	10.	11.	19.	23.	16.	16.	19.	17.	36.	37.	12.	12.	
12.	24.	14.	11.	11.	14.	11.	12.	12.	13.	13.	16.	17.	9.	10.	11.	11.	25.	15.	19.	29.	10.	11.	19.	23.	16.	15.	18.	17.	35.	36.	12.	12.	
12.	24.	14.	11.	11.	14.	12.	12.	12.	13.	13.	16.	16.	9.	9.	11.	11.	26.	15.	19.	27.	11.	11.	19.	23.	15.	15.	18.	18.	33.	36.	12.	12.	
12.	24.	15.	11.	11.	15.	11.	13.	14.	13.	13.	16.	20.	9.	10.	11.	11.	25.	15.	19.	27.	11.	10.	19.	23.	15.	15.	17.	17.	35.	38.	12.	12.	
12.	25.	14.	10.	11.	12.	12.	11.	13.	13.	13.	16.	17.	10.	10.	11.	11.	26.	15.	19.	29.	11.	11.	20.	23.	15.	15.	18.	17.	34.	38.	13.	12.	
12.	25.	14.	10.	11.	14.	11.	12.	12.	13.	13.	17.	16.	9.	10.	11.	11.	24.	16.	19.	28.	10.	11.	19.	23.	16.	16.	18.	17.	35.	39.	12.	12.	
12.	25.	14.	10.	12.	14.	11.	12.	11.	13.	13.	16.	17.	9.	10.	11.	11.	24.	15.	20.	29.	10.	10.	17.	23.	18.	17.	18.	16.	34.	35.	12.	11.	
12.	25.	14.	10.	13.	14.	11.	12.	11.	13.	13.	16.	18.	9.	9.	11.	11.	24.	15.	19.	31.	11.	11.	17.	23.	15.	16.	19.	18.	37.	37.	12.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	12.	14.	14.	16.	17.	9.	10.	11.	11.	25.	15.	19.	30.	10.	10.	19.	23.	15.	15.	18.	18.	36.	37.	14.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	11.	13.	13.	16.	16.	9.	9.	11.	11.	25.	15.	19.	28.	10.	11.	19.	23.	15.	16.	20.	19.	36.	37.	13.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	12.	13.	13.	16.	15.	9.	10.	11.	11.	24.	15.	18.	31.	10.	10.	19.	23.	16.	16.	17.	17.	36.	37.	12.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	12.	13.	13.	16.	17.	9.	10.	11.	11.	25.	15.	19.	28.	10.	11.	18.	23.	15.	15.	20.	18.	34.	38.	12.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	12.	13.	13.	16.	18.	9.	10.	11.	11.	25.	15.	20.	29.	10.	11.	17.	23.	16.	16.	18.	17.	37.	38.	12.	12.	
12.	25.	15.	11.	11.	14.	11.	12.	12.	13.	13.	16.	16.	9.	10.	11.	11.	25.	14.	19.	29.	12.	11.	19.	23.	17.	16.	18.	17.	35.	39.	12.	11.	
12.	25.	15.	11.	11.	14.	11.	13.	13.	13.	13.	17.	20.	9.	10.	11.	11.	25.	15.	19.	27.	11.	11.	19.	23.	15.	15.	17.	17.	35.	38.	12.	12.	
12.	25.	15.	11.	12.	15.	11.	12.	12.	13.	13.	16.	16.	9.	10.	11.	11.	24.	15.	18.	31.	10.	10.	17.	24.	15.	17.	18.	17.	34.	36.	12.	12.	
12.	25.	14.	11.	11.	14.	11.	12.	12.	13.	13.	16.	17.	9.	10.	11.	11.	25.	15.	19.	30.	10.	10.	17.	23.	15.	16.	17.	17.	36.	37.	12.	12.	
12.	23.	14.	10.	11.	15.	12.	12.	12.	13.	14.	15.	16.	9.	10.	11.	11.	25.	15.	19.	28.	11.	12.	19.	23.	15.	15.	17.	17.	35.	37.	12.	12.	
12.	24.	14.	10.	11.	12.	12.	12.	12.	13.	13.	16.	16.	9.	10.	11.	11.	23.	15.	18.	29.	11.	12.	19.	23.	17.	17.	17.	17.	36.	37.	12.	12.	
12.	24.	14.	10.	11.	14.	12.	12.	11.	13.	14.	16.	17.	9.	10.	11.	11.	25.	15.	19.	29.	11.	12.	19.	23.	17.	15.	17.	17.	38.	39.	12.	12.	
12.	24.	14.	10.	11.	14.	12.	12.	13.	12.	14.	16.	16.	9.	10.	11.	11.	26.	15.	19.	28.	11.	12.	19.	23.	15.	16.	19.	17.	37.	39.	12.	12.	
12.	24.	14.	10.	11.	14.	12.	12.	13.	12.	14.	16.	16.	9.	10.	11.	11.	25.	15.	19.	28.	11.	12.	19.	23.	15.	16.	19.	17.	37.	39.	12.	12.	
12.	24.	14.	11.	11.	14.	12.	12.	12.	13.	13.	16.	17.	9.	9.	11.	11.	25.	15.															

12	24	15	11	13	14	12	12	13	13	13	16	16	8	10	11	11	25	15	19	28	12	11	19	21	16	16	18	16	35	37	12	12	
12	25	13	11	11	14	12	12	12	13	13	17	16	7	10	11	11	25	15	20	30	11	12	19	23	15	15	20	16	36	39	12	12	
12	25	14	11	11	14	12	11	12	13	13	16	15	9	10	11	11	24	15	19	29	11	12	19	23	15	15	17	16	35	37	11	12	
12	25	14	11	11	14	12	12	12	14	13	18	17	9	9	11	11	26	15	19	28	11	11	19	23	15	15	18	14	37	38	12	12	
12	25	14	11	12	15	12	12	12	13	13	16	16	9	10	11	11	25	15	20	28	11	11	23	23	16	16	17	17	35	35	12	12	
12	25	14	11	12	15	12	12	12	13	13	16	16	9	10	11	11	25	15	20	28	11	11	23	23	16	16	18	17	35	35	12	12	
12	25	14	11	12	15	12	12	11	13	13	17	16	9	10	11	11	25	15	19	28	11	11	19	23	15	15	17	18	37	37	12	12	
13	24	14	10	11	14	12	12	13	13	13	19	16	10	10	11	11	25	15	19	29	11	11	22	23	15	15	18	15	34	36	12	12	
13	24	14	10	11	14	12	12	13	14	13	16	17	9	10	11	11	25	15	19	28	11	11	19	23	15	15	14	34	37	12	12		
13	24	14	10	11	16	13	12	12	14	13	16	17	9	9	11	11	25	15	19	29	11	11	19	23	15	15	19	15	34	38	11	12	
12	23	14	10	11	14	12	12	13	13	13	16	16	9	10	11	11	25	15	19	30	11	11	19	23	15	16	17	17	37	38	12	12	
12	23	14	11	11	14	12	12	13	13	13	15	16	9	10	11	11	24	14	19	31	12	11	19	23	15	15	18	16	37	37	12	12	
12	23	14	11	11	15	12	12	11	13	13	16	16	9	10	11	11	24	14	19	32	11	12	19	22	16	14	20	17	37	38	12	12	
12	24	14	10	11	13	12	12	12	13	13	16	17	9	10	11	11	25	15	19	27	11	11	19	23	15	15	19	19	35	41	12	12	
12	24	14	11	11	12	12	12	12	13	13	17	17	9	10	11	11	25	15	19	31	11	11	19	23	15	16	20	18	36	39	12	12	
12	24	14	11	11	14	12	12	12	13	13	16	15	9	9	11	11	25	14	19	30	11	10	19	23	16	17	18	39	39	11	12		
12	24	14	11	11	14	12	12	12	13	13	17	16	9	10	11	11	25	15	19	29	11	11	19	23	15	16	19	18	38	38	12	12	
12	24	14	11	11	15	12	12	12	14	13	17	17	9	10	11	11	25	14	20	29	12	11	19	23	15	16	18	19	37	38	12	12	
12	24	14	11	12	14	12	12	11	13	13	17	18	9	10	11	11	25	16	19	30	10	10	12	19	23	15	16	19	18	37	38	12	12
12	24	14	11	12	14	12	12	13	13	13	16	18	9	10	11	11	25	15	19	32	11	12	19	23	15	15	19	17	35	38	12	12	
12	24	14	11	12	14	12	13	11	12	13	16	16	9	10	11	11	25	15	19	29	11	11	19	23	15	16	17	17	37	39	12	12	
12	24	15	10	12	13	12	12	11	13	13	15	17	9	9	11	11	25	15	19	29	11	11	19	23	15	18	18	17	36	37	12	12	
12	24	15	11	11	11	12	12	13	14	13	16	16	9	9	11	11	25	15	19	30	11	12	19	23	15	16	17	17	37	37	13	12	
12	25	14	11	11	14	12	12	12	13	13	16	16	9	9	11	11	25	14	18	27	11	11	19	23	15	18	18	18	37	37	12	12	
12	25	14	11	11	14	12	12	13	13	13	16	16	9	9	11	11	25	15	18	30	11	12	19	23	15	15	18	19	36	37	12	12	
12	25	14	11	11	14	12	12	14	13	14	16	16	9	9	11	11	26	15	19	32	11	12	19	24	15	15	18	17	37	37	12	12	
12	25	14	11	12	14	12	12	14	13	13	16	16	9	10	11	11	25	16	20	29	11	12	19	23	16	15	16	17	36	37	12	12	
12	25	14	11	12	14	12	12	13	13	16	16	9	10	11	11	25	15	19	29	11	11	19	23	15	15	18	15	18	36	36	13	12	
12	24	14	10	11	14	12	12	13	13	13	16	18	9	10	11	11	25	15	19	29	10	11	18	23	15	17	19	16	35	36	13	12	
12	24	14	10	11	15	12	12	11	13	13	16	17	9	9	11	11	24	16	19	31	11	11	19	21	17	15	18	17	36	36	12	12	
12	24	14	10	11	15	12	12	12	13	13	16	16	9	9	11	11	24	15	19	30	11	11	19	21	16	15	18	17	36	37	12	12	
12	24	14	10	11	15	12	12	13	13	16	17	9	9	11	11	24	15	19	32	11	11	19	21	17	15	18	17	35	36	12	12		
12	24	14	10	11	15	12	12	15	13	13	16	17	9	10	11	11	25	15	19	33	10	11	19	23	15	15	19	17	36	37	12	12	
12	24	14	11	11	14	12	12	12	13	14	17	17	9	9	11	11	25	15	19	29	11	11	18	23	16	15	18	17	36	37	11	12	
12	24	14	11	11	14	12	12	14	13	13	16	17	9	10	11	11	25	14	19	29	10	11	19	23	16	17	17	37	35	40	12	12	
12	24	14	11	11	15	12	12	12	13	13	16	16	9	10	11	11	26	15	19	31	11	11	19	23	15	16	19	18	35	37	12	12	
12	24	15	11	11	14	12	12	12	13	16	17	9	10	11	11	24	15	20	30	11	12	19	23	15	16	18	17	35	36	12	13		
12	25	13	10	12	14	12	12	12	13	12	16	18	9	10	11	11	25	16	20	29	11	12	19	23	16	15	16	17	36	37	12	12	
12	25	14	11	12	14	12	12	12	13	16	16	9	10	11	11	25	15	19	29	10	11	19	21	16	15	18	18	38	39	12	12		
12	23	14	11	11	15	12	12	12	13	13	16	16	9	9	11	11	25	15	19	29	11	13	19	23	16	16	17	17	36	37	12	12	
12	24	14	11	11	14	12	12	11	13	14	16	16	9	10	11	11	25	15	19	29	11	12	19	23	16	16	17	17	37	37	12	12	
12	24	14	11	12	13	12	12	13	14	13	16	17	9	10	11	11	25	15	19	30	10	10	19	23	15	14	18	17	36	37	12	12	
12	23	14	11	11	14	12	12	11	13	13	16	18	9	10	11	11	25	15	19	29	11	11	19	23	16	14	17	17	37	37	12	12	
14	24	14	10	11	14	12	12	12	13	13	16	17	9	10	11	11	25	15	19	28	11	11	19	23	16	14	18	17	36	36	12	12	
12	23	14	11	11	14	12	12	12	13	16	17	9	10	11	11	25	15	19	28	11	11	19	23	16	14	18	17	36	36	12	12		
12	23	14	11	11	15	12	12	12	13	16	16	9	10	11	11	25	16	19	29	11	12	19	23	16	16	17	17	37	37	12	12		
12	23	14	11	12	14	12	12	12	13	14	15	16	9	10	11	11	25	16	19	29	11	12	19	23	16	16	17	17	37	37	12	12	
12	23	14	11	12	14	12	12	11	14	13	16	16	9	10	11	11	25	15	19	29	11	12	19	23	16	15	17	17	37	40	12	12	
12	24	14	10	11	12	12	12	12	13	14	15	16	9	10	11	11	23	15	19	29	10	12	19	23	17	16	18	17	37	38	10	12	
12	24	14	10	11	15	12	12	13	13	13	16	18	9	10	11	11	25	15	19	29	11	12	19	19	15	14	20	16	35	38	12	12	
12	24	14	11	11	14	12	12	12	13	17	15	9	9	11	11	26	15	19	28	11	11	19	23	16	15	16	20	17	34	39	13	12	
12	24	14	11	11	14	12	12	12	13	17	15	9	10	11	11	24	14	20	29	11	11	19	22	15	15	18	16	35	38	12			


```

12. 25. 14. 11. 11. 15. 12. 12. 13. 13. 13. 17. 15. 9. 9. 11. 11. 26. 14. 19. 29. 10. 12. 19. 23. 16. 15. 17. 17. 37. 37. 12. 12.
12. 25. 14. 11. 11. 16. 12. 12. 13. 13. 13. 16. 17. 9. 9. 11. 11. 25. 15. 19. 29. 11. 12. 19. 23. 16. 16. 19. 17. 37. 36. 12. 12.
12. 25. 15. 11. 11. 15. 12. 12. 13. 13. 13. 17. 15. 9. 9. 11. 11. 26. 14. 19. 29. 10. 12. 19. 23. 16. 15. 17. 17. 37. 36. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 12. 14. 13. 17. 19. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 16. 15. 18. 18. 34. 36. 12. 12.
13. 24. 14. 11. 11. 16. 12. 12. 12. 14. 13. 17. 19. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 16. 15. 17. 17. 36. 37. 12. 12.
13. 24. 14. 11. 11. 12. 15. 12. 12. 12. 14. 13. 16. 15. 9. 10. 11. 11. 25. 15. 20. 28. 12. 13. 19. 23. 17. 16. 18. 15. 36. 36. 12. 12.
13. 24. 14. 12. 11. 15. 12. 12. 12. 13. 13. 17. 17. 9. 10. 11. 11. 26. 15. 19. 29. 12. 11. 19. 23. 15. 15. 16. 16. 36. 36. 12. 12.
13. 25. 13. 11. 11. 14. 12. 12. 12. 13. 13. 13. 17. 16. 7. 10. 11. 11. 25. 15. 20. 30. 11. 13. 19. 23. 15. 15. 21. 16. 36. 41. 12. 12.
12. 23. 14. 10. 11. 15. 12. 12. 12. 13. 13. 15. 17. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 23. 16. 15. 16. 17. 37. 39. 12. 12.
12. 23. 14. 10. 11. 15. 12. 12. 12. 13. 13. 14. 15. 18. 9. 10. 11. 11. 24. 15. 19. 29. 10. 11. 19. 22. 16. 15. 18. 16. 35. 39. 12. 12.
12. 23. 14. 11. 11. 15. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 27. 10. 12. 19. 22. 16. 16. 17. 17. 36. 36. 12. 12.
12. 23. 14. 11. 11. 15. 12. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 25. 15. 19. 27. 11. 12. 19. 23. 15. 16. 16. 17. 36. 39. 11. 12.
12. 23. 14. 11. 11. 16. 12. 12. 12. 13. 15. 15. 17. 9. 10. 11. 11. 25. 15. 19. 27. 10. 11. 19. 22. 16. 16. 18. 18. 36. 39. 11. 12.
12. 23. 14. 11. 11. 14. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 23. 16. 15. 18. 15. 36. 36. 12. 12.
12. 23. 15. 11. 11. 14. 12. 12. 12. 12. 14. 15. 16. 9. 9. 11. 11. 25. 15. 19. 30. 8. 11. 19. 23. 15. 15. 18. 17. 34. 34. 12. 12.
12. 24. 14. 10. 11. 14. 12. 12. 12. 12. 14. 15. 18. 9. 10. 11. 11. 25. 14. 19. 29. 12. 12. 19. 23. 15. 17. 19. 17. 36. 37. 13. 12.
12. 24. 14. 10. 11. 14. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 31. 10. 11. 19. 23. 15. 16. 17. 16. 38. 39. 12. 12.
12. 24. 14. 10. 11. 14. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 31. 10. 11. 19. 23. 15. 16. 17. 17. 38. 40. 12. 12.
12. 24. 14. 10. 11. 14. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 26. 15. 19. 30. 11. 11. 19. 23. 17. 16. 17. 17. 37. 37. 11. 12.
12. 24. 14. 10. 11. 14. 12. 12. 13. 14. 15. 18. 9. 10. 11. 11. 25. 15. 19. 29. 11. 12. 19. 23. 15. 16. 17. 14. 36. 36. 13. 12.
12. 24. 14. 10. 11. 14. 12. 12. 14. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 12. 19. 23. 15. 14. 18. 17. 36. 36. 12. 12.
12. 24. 14. 10. 11. 15. 12. 12. 11. 13. 14. 15. 16. 9. 11. 11. 11. 25. 15. 19. 29. 12. 12. 19. 23. 15. 15. 18. 17. 36. 39. 12. 12.
12. 24. 14. 10. 11. 15. 12. 12. 12. 13. 14. 15. 15. 9. 9. 11. 11. 25. 14. 19. 29. 11. 12. 19. 23. 15. 16. 19. 17. 34. 38. 12. 12.
12. 24. 14. 10. 11. 15. 12. 12. 12. 13. 14. 15. 17. 9. 9. 11. 11. 25. 15. 19. 28. 12. 12. 19. 23. 14. 16. 16. 19. 37. 38. 11. 12.
12. 24. 14. 10. 11. 15. 12. 12. 13. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 12. 19. 23. 15. 16. 17. 17. 36. 39. 12. 12.
12. 24. 14. 10. 11. 15. 12. 12. 13. 13. 14. 15. 18. 9. 9. 11. 11. 25. 15. 19. 28. 10. 11. 19. 23. 15. 16. 16. 17. 36. 36. 12. 12.
12. 24. 14. 10. 11. 16. 12. 12. 12. 13. 14. 15. 19. 9. 10. 11. 11. 25. 14. 19. 26. 10. 12. 19. 23. 16. 16. 17. 17. 38. 39. 10. 12.
12. 24. 14. 11. 10. 14. 12. 12. 12. 14. 14. 15. 18. 9. 10. 11. 11. 25. 15. 19. 30. 11. 11. 19. 21. 17. 15. 16. 17. 33. 36. 11. 12.
12. 24. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 16. 9. 9. 11. 11. 25. 15. 19. 28. 12. 13. 19. 23. 15. 15. 18. 17. 37. 39. 12. 12.
12. 24. 14. 11. 11. 14. 12. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 25. 15. 19. 27. 10. 12. 19. 23. 15. 16. 17. 18. 36. 37. 11. 12.
12. 24. 14. 11. 11. 14. 12. 12. 12. 13. 14. 15. 17. 9. 10. 12. 11. 25. 15. 19. 30. 11. 11. 19. 23. 16. 17. 17. 37. 40. 12. 12.
12. 24. 14. 11. 11. 14. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 22. 18. 16. 17. 17. 34. 37. 12. 12.
12. 24. 14. 11. 11. 14. 12. 12. 13. 14. 16. 14. 9. 9. 11. 11. 25. 15. 19. 27. 10. 12. 19. 23. 15. 16. 19. 17. 37. 38. 12. 12.
12. 24. 14. 11. 11. 15. 12. 12. 11. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 28. 9. 11. 19. 22. 15. 17. 17. 18. 35. 38. 11. 12.
12. 24. 14. 11. 11. 15. 12. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 25. 15. 19. 27. 10. 12. 19. 23. 15. 16. 18. 18. 36. 37. 11. 12.
12. 24. 14. 11. 11. 15. 12. 12. 12. 13. 14. 15. 18. 9. 9. 11. 11. 25. 15. 19. 30. 11. 10. 19. 23. 16. 16. 18. 17. 34. 34. 11. 12.
12. 24. 14. 11. 11. 16. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 17. 15. 18. 18. 36. 36. 12. 12.
12. 24. 14. 11. 12. 14. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 17. 16. 17. 35. 36. 12. 12.
12. 24. 14. 11. 12. 14. 12. 12. 13. 13. 14. 15. 16. 9. 9. 11. 11. 26. 15. 19. 29. 10. 11. 19. 23. 17. 15. 17. 18. 36. 36. 12. 12.
12. 24. 14. 11. 13. 14. 12. 12. 13. 13. 14. 15. 18. 9. 10. 11. 11. 26. 15. 19. 28. 12. 11. 19. 23. 17. 16. 16. 18. 36. 37. 11. 12.
12. 25. 14. 11. 11. 13. 12. 12. 12. 12. 14. 16. 18. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 23. 15. 16. 17. 16. 37. 38. 12. 12.
12. 25. 14. 11. 11. 14. 12. 12. 12. 13. 14. 15. 15. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 15. 16. 18. 17. 36. 36. 12. 12.
12. 25. 14. 11. 12. 14. 12. 12. 12. 13. 16. 16. 9. 10. 11. 11. 25. 15. 20. 28. 11. 11. 19. 23. 15. 15. 19. 18. 36. 37. 13. 12.
12. 25. 14. 11. 12. 14. 12. 12. 12. 13. 14. 15. 16. 9. 10. 11. 11. 25. 15. 19. 29. 11. 12. 19. 24. 15. 16. 18. 17. 36. 37. 12. 12.
12. 26. 14. 10. 12. 14. 12. 12. 12. 13. 12. 15. 15. 17. 9. 10. 11. 11. 25. 15. 19. 30. 11. 11. 19. 23. 15. 16. 19. 18. 36. 37. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 12. 14. 14. 16. 9. 9. 11. 11. 26. 15. 19. 29. 11. 12. 19. 23. 15. 16. 18. 17. 38. 38. 12. 12.
13. 25. 14. 10. 11. 15. 12. 12. 12. 13. 14. 15. 17. 9. 10. 11. 11. 26. 15. 19. 28. 12. 12. 19. 23. 15. 16. 17. 19. 35. 38. 12. 12.
12. 23. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 28. 11. 19. 23. 16. 16. 19. 16. 35. 36. 12. 12.
12. 23. 14. 11. 12. 14. 12. 12. 12. 13. 13. 16. 14. 9. 10. 11. 11. 25. 15. 19. 32. 11. 11. 19. 23. 16. 15. 17. 17. 35. 41. 12. 12.
12. 24. 13. 11. 12. 14. 12. 12. 13. 13. 13. 17. 16. 9. 10. 11. 11. 25. 15. 19. 29. 10. 11. 19. 23. 15. 16. 20. 18. 36. 36. 13. 12.
12. 24. 14. 10. 11. 13. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 31. 12. 12. 19. 23. 16. 15. 18. 19. 35. 36. 12. 12.
12. 24. 14. 10. 11. 14. 12. 12. 12. 13. 13. 17. 16. 9. 10. 11. 11. 25. 15. 19. 28. 11. 11. 19. 23. 15. 16. 19. 18. 37. 39. 14. 12.
12. 24. 14. 10. 11. 14. 12. 12. 12. 13. 13. 17. 18. 9. 10. 11. 11. 25. 15. 19. 32. 10. 11. 19. 23. 15. 16. 19. 18. 37. 40. 11. 12.
12. 24. 14. 10. 11. 14. 12. 12. 13. 14. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 22. 15. 13. 18. 17. 36. 37. 12. 12.
12. 24. 14. 10. 11. 14. 12. 12. 13. 13. 13. 16. 15. 9. 10. 11. 11. 26. 15. 19. 31. 10. 11. 19. 23. 15. 17. 21. 18. 37. 40. 12. 12.
12. 24. 14. 10. 11. 15. 12. 12. 12. 13. 13. 17. 16. 9. 10. 11. 11. 25. 15. 19. 28. 11. 11. 19. 23. 15. 16. 19. 17. 36. 38. 13. 12.
12. 24. 14. 11. 11. 13. 12. 12. 12. 13. 13. 16. 16. 9. 10. 11. 11. 25. 15. 19. 31. 11. 11. 19. 23. 16. 14. 19. 18. 35. 37. 11. 12.
12. 24. 14. 11. 11. 14. 12. 12. 12. 12. 13. 16. 15. 9. 9. 11. 11. 25. 14. 19. 29. 11. 10. 19. 23. 16. 17. 18. 18. 39. 39. 11. 12.
12. 24. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 30. 10. 11. 19. 23. 16. 16. 17. 17. 36. 36. 12. 12.
12. 24. 14. 11. 11. 16. 12. 12. 12. 13. 13. 16. 16. 9. 10. 11. 11. 25. 15. 19. 30. 10. 11. 19. 23. 15. 15. 20. 17. 37. 38. 11. 12.
12. 24. 14. 11. 11. 15. 12. 12. 12. 13. 13. 15. 16. 6. 10. 11. 11. 26. 15. 19. 29. 11. 12. 19. 23. 16. 15. 18. 18. 37. 38. 12. 12.
12. 24. 14. 12. 11. 14. 12. 12. 12. 13. 13. 16. 16. 9. 9. 11. 11. 25. 15. 19. 27. 11. 12. 19. 23. 15. 15. 17. 16. 37. 39. 12. 12.
12. 24. 15. 11. 11. 14. 12. 12. 12. 14. 14. 16. 17. 9. 10. 11. 11. 22. 15. 19. 28. 10. 12. 19. 23. 16. 16. 18. 17. 37. 37. 11. 12.
12. 25. 14. 11. 11. 13. 11. 12. 12. 14. 13. 16. 17. 9. 10. 11. 11. 25. 15. 18. 30. 11. 11. 19. 23. 16. 16. 18. 17. 35. 42. 13. 12.
12. 25. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 30. 11. 12. 19. 23. 16. 17. 19. 18. 35. 36. 12. 12.
12. 25. 14. 11. 10. 14. 12. 12. 13. 14. 13. 17. 15. 9. 10. 11. 11. 25. 15. 18. 29. 10. 12. 19. 23. 15. 16. 18. 17. 34. 40. 13. 12.
13. 23. 14. 11. 11. 15. 13. 12. 12. 12. 13. 13. 16. 18. 8. 10. 10. 11. 25. 16. 19. 31. 11. 11. 19. 23. 16. 16. 19. 17. 37. 37. 10. 12.
13. 24. 14. 10. 12. 14. 13. 12. 12. 13. 13. 17. 17. 9. 10. 11. 11. 24. 15. 19. 28. 11. 11. 19. 22. 15. 16. 17. 17. 37. 39. 12. 12.
13. 24. 14. 11. 11. 14. 13. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 28. 11. 11. 19. 23. 16. 15. 17. 18. 37. 38. 12. 12.
13. 24. 14. 11. 11. 14. 13. 12. 12. 13. 13. 17. 17. 9. 10. 11. 11. 25. 15. 20. 30. 11. 10. 19. 23. 16. 16. 17. 17. 36. 38. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 27. 15. 19. 29. 11. 9. 19. 23. 16. 15. 19. 17. 35. 38. 12. 12.
13. 25. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 16. 9. 9. 11. 11. 25. 15. 19. 30. 11. 12. 19. 23. 15. 16. 20. 18. 37. 37. 12. 12.
13. 25. 14. 11. 11. 14. 13. 12. 12. 13. 13. 16. 17. 9. 9. 11. 11. 25. 15. 19. 29. 11. 11. 19. 23. 16. 15. 17. 18. 36. 38. 12. 12.
12. 23. 14. 11. 10. 14. 12. 12. 12. 13. 13. 16. 19. 9. 10. 11. 11. 25. 15. 18. 29. 11. 11. 19. 23. 15. 15. 16. 15. 36. 38. 11. 12.
13. 23. 13. 10. 11. 14. 12. 12. 11. 12. 13. 16. 17. 10. 10. 11. 11. 25. 15. 19. 29. 12. 11. 19. 23. 16. 15. 18. 18. 35. 35. 12. 12.
13. 23. 14. 11. 11. 14. 12. 12. 11. 13. 14. 16. 17. 9. 10. 11. 11. 25. 15. 19. 31. 10. 9. 19. 23. 16. 14. 19. 18. 37. 38. 12. 12.
13. 24. 14. 10. 11. 15. 12. 12. 13. 14. 13. 16. 18. 9. 10. 11. 11. 25. 15. 19. 28. 11. 11. 19. 23. 16. 15. 15. 16. 37. 37. 12. 12.
13. 24. 14. 11. 10. 14. 12. 12. 14. 13. 13. 16. 18. 9. 10. 11. 11. 25. 15. 19. 29. 10. 11. 19. 21. 16. 16. 17. 17. 35. 38. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 12. 13. 13. 15. 16. 9. 10. 11. 11. 24. 15. 19. 28. 11. 11. 19. 23. 16. 15. 18. 18. 37. 37. 10. 13.
13. 24. 14. 11. 11. 14. 12. 12. 12. 13. 14. 16. 18. 9. 10. 11. 11. 25. 15. 19. 29. 11. 11. 19. 23. 16. 16. 20. 17. 37. 37. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 13. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 27. 11. 11. 19. 23. 16. 15. 20. 17. 36. 38. 13. 12.
13. 24. 14. 11. 11. 14. 12. 12. 13. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 29. 11. 10. 19. 23. 16. 15. 17. 19. 36. 40. 12. 12.
13. 24. 14. 11. 11. 14. 12. 12. 13. 14. 12. 15. 15. 9. 10. 11. 11. 26. 15. 19. 29. 11. 12. 19. 23. 16. 15. 19. 18. 36. 37. 10. 13.
13. 24. 14. 11. 11. 14. 12. 12. 13. 13. 13. 16. 16. 9. 9. 11. 11. 25. 15. 19. 31. 11. 10. 19. 23. 16. 15. 17. 17. 36. 39. 12. 12.
13. 25. 14. 10. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 26. 15. 19. 29. 11. 11. 19. 23. 15. 15. 18. 17. 36. 38. 12. 12.
13. 25. 14. 11. 11. 14. 12. 12. 13. 13. 16. 9. 9. 11. 11. 25. 15. 19. 30. 11. 12. 19. 23. 15. 16. 19. 18. 37. 39. 12. 12.
13. 25. 14. 11. 11. 15. 12. 12. 12. 13. 13. 16. 18. 9. 10. 11. 11. 25. 14. 19. 29. 10. 11. 19. 23. 15. 14. 17. 17. 36. 40. 12. 12.
13. 25. 14. 11. 11. 15. 12. 12. 12. 13. 13. 16. 18. 9. 10. 11. 11. 25. 14. 19. 29. 10. 11. 19. 23. 15. 14. 17. 17. 36. 40. 12. 12.
14. 23. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 10. 11. 11. 25. 15. 19. 32. 11. 11. 19. 23. 15. 15. 17. 19. 37. 39. 12. 12.
14. 23. 14. 11. 11. 14. 12. 12. 12. 13. 13. 16. 17. 9. 11. 11. 11. 25. 14. 19. 29. 11. 11. 19. 23. 16. 15. 17. 19. 35. 35. 12. 12.

```

NN = Length[*δδ*]