CSE 280A
Assignment 3
List 1
Andrey Bzikadze

① I used msms simulator for this task.

Two scenarios: (1) $\alpha = 0$ // constant population size
(2) $\alpha = 10$ // expon. growth.

In msms I used parameter -G for that.

Each sample is the size of 100. Totally 1000 samples
Other params are from the task formulation:

$\theta = 40 \quad N_0 = 10^6$.

List 2 shows comparison of AFS for both cases summed up from all samples (parameter -oAFS only Summary) // trimmed first 25 pos.
List 3 shows the same but with normalized on ~~total sum of~~ total # of mutations.
List 4 shows estimate of θ with the formula $E m_i = \frac{\theta}{i}$

=> $\boxed{\hat{\theta}_i = \frac{\hat{m}_i \cdot i}{n}}$. For $N = const$ $\hat{\theta}_i$ is an unbiased estimate ($E\hat{\theta}_i = \theta$) for any i. The variance ($Var(\hat{\theta}_i)$) rises with i just because $m_i \downarrow$ with $i \to \infty$. For N-exp. growing $\hat{\theta}_i$ is clearly biased and underestimating the real θ.

Why this happens? For $N = const$ $T_k \sim Geom\left(\frac{\binom{k}{2}}{N}\right)$

=> $ET_k = \frac{N}{\binom{k}{2}}$ => $ET_k \downarrow$ as k increases.

=> more mutations are introduced in early edges (as is proport. to the length)
=> more mutations are inherited.
(comparing to N-exp., as we will see)

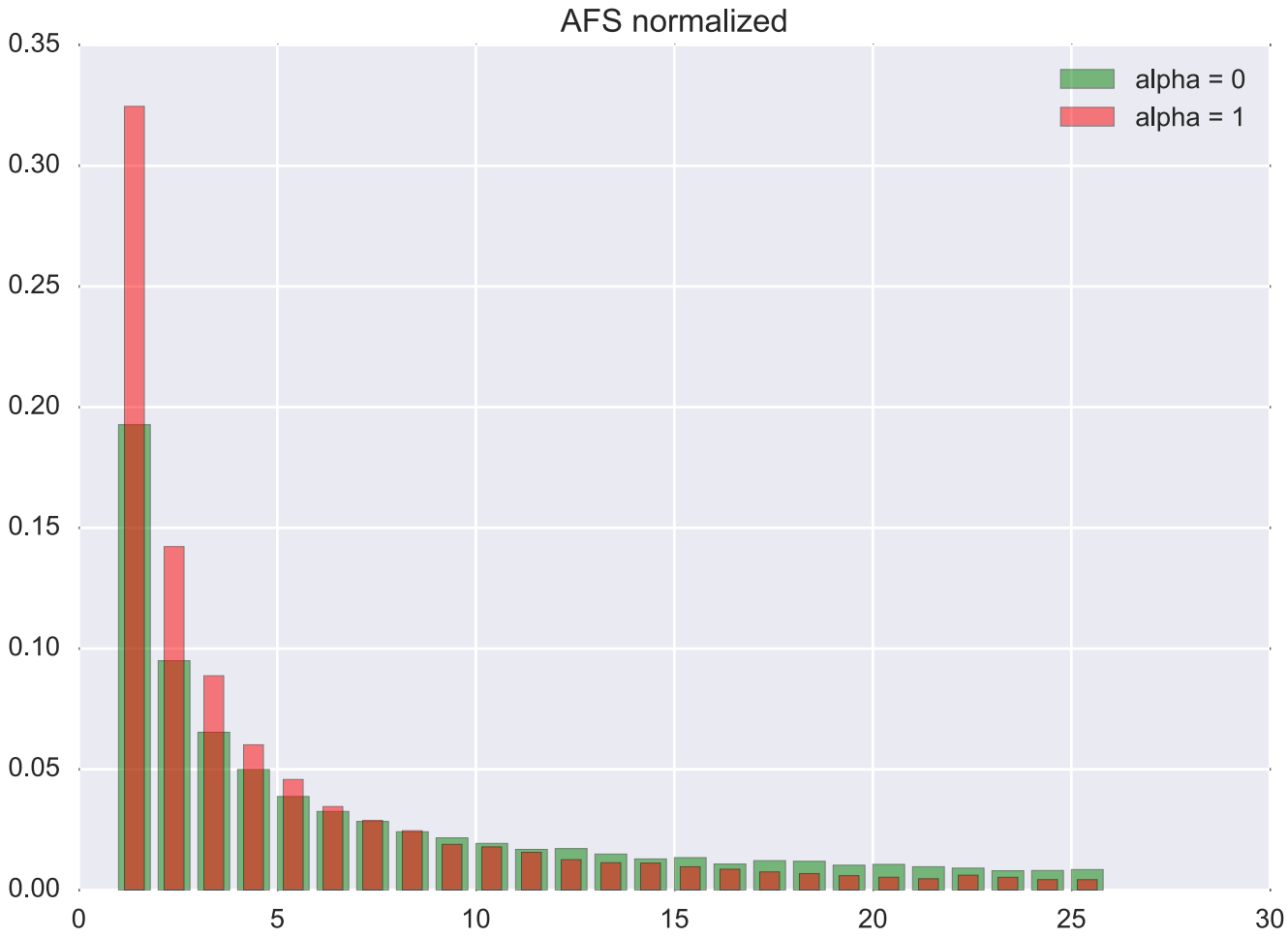
For N-expon. T_k have a more complex distribution that I described in previous assignment and will briefly do here:

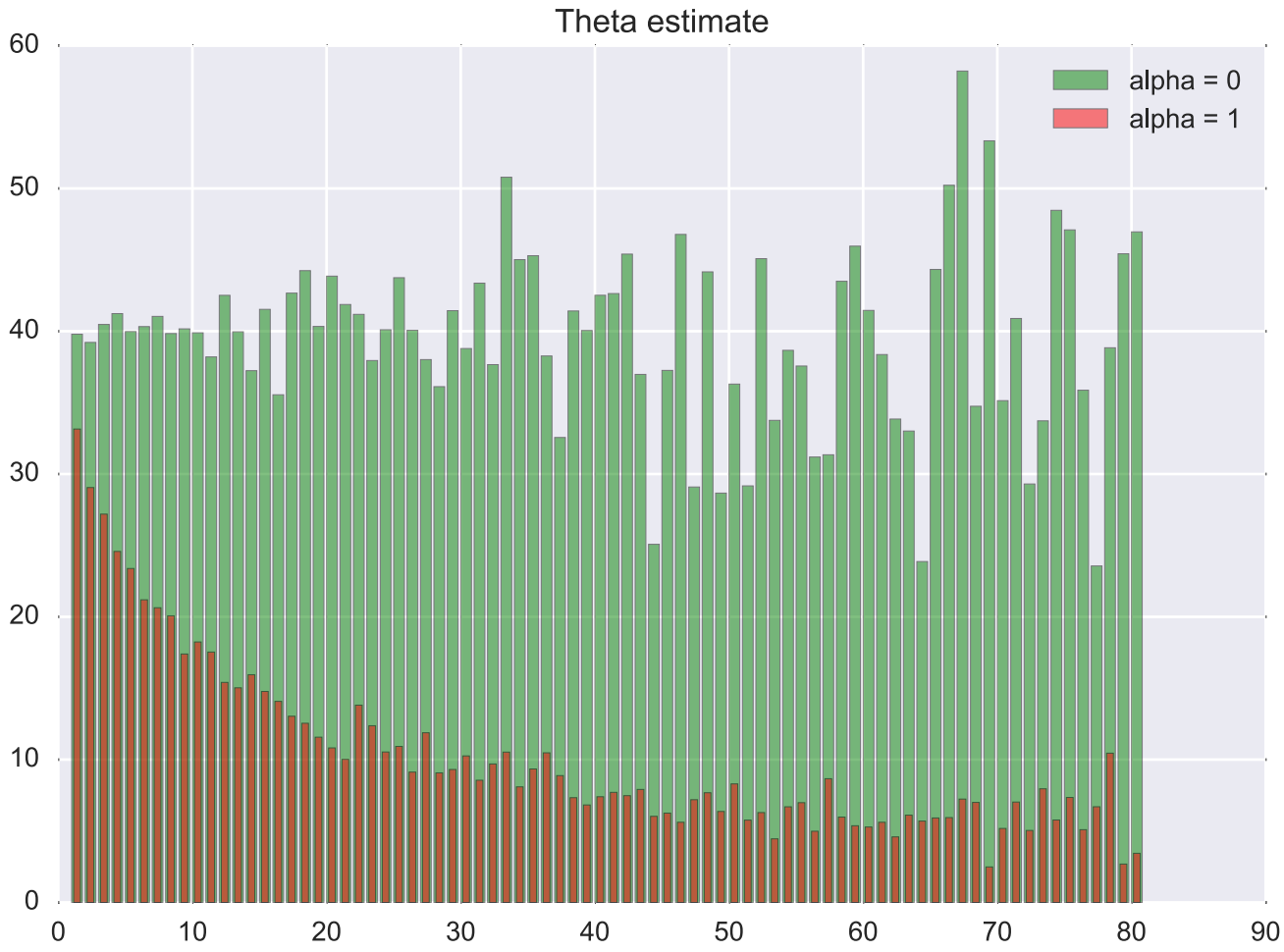
$$P(T_k = l+1) = \frac{\binom{k}{2}}{N_k} e^{-\alpha l} \prod_{s=1}^{l} \left(1 - \frac{\binom{k}{2}}{N_k} e^{-\alpha s}\right) \quad \text{where } N_k = N_0 e^{-\alpha k}.$$

We can estimate $ET_k \approx \frac{N_k}{\binom{k}{2}}$ where N_k grows exponentially and $\binom{k}{2}$ grows polynomially

=> $ET_k \uparrow$ with k increasing. =>
=> more mutations are introduced in later edges =>
=> less (compared to N=const) mutations are inherited.


AFS unnormalized
alpha = 0
alpha = 1
40000
35000
30000
25000
20000
15000
10000
5000
0
0
5
10
15
20
25
30


AFS normalized
alpha = 0
alpha = 1
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30

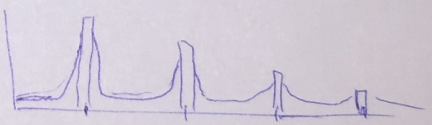

Theta estimate
alpha = 0
alpha = 1
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90

CSE 280 A
Assignment 3
List 5
Audrey Bzikadze

② We need a test for selection based on statistics $D = \hat{\theta}_1 - \hat{\theta}_2$ where $\hat{\theta}_1, \hat{\theta}_2$ are two different estimates of θ. We'd like to have a Z-score $Z = D/std(D) \underset{n\to\infty}{\sim} \mathcal{N}(0;1)$. under null hypothesis that there is no selection, => pvalue = $2\min(\Phi^{-1}(z), 1-\Phi^{-1}(z))$.

1) I propose $\hat{\theta}_1$ - a Tajima's k. From assignment 2.5 we know that it looks like $k = \sum_{j=1}^{n} f_j(1-f_j)$, where f_j is the frequency of mutation allele at site j. I'm not sure that $Ek = \theta$ (exactly), but I think it is possible to estimate the possible bias with simulations and account for it.

2) For $\hat{\theta}_2$ I'd propose a simple method to estimate coverage of a distinct indiv. First, suppose that $n_i = n_j \ \forall i,j$. and reads are error-free. We as usual denote m_i as # mutations with exactly i ones (reads)
=> $\bar{m}_i$ (from data) : $\bar{m}_i = 0 \ \forall i < C$ where C is coverage of a distinct indiv.
If the reads have errors and the coverage of distinct individuals is not exactly identical (but close) then $\bar{m}_C \gg \bar{m}_i \ \forall i < C$ and AFS could look like this (under no selection)



The same when $n_i \neq n_j$ exactly, but close.
=> We can define estimates: $\hat{\theta}_i^{(2)} = i \cdot C \cdot \bar{m}_{iC}$
It is known that $E\hat{\theta}_i^{(2)} = \theta_2$. // $iC \le n-1$ => $i \le \left\lfloor \frac{n-1}{C} \right\rfloor$

Finally, Define $\hat{\theta}_2 := \frac{1}{\left\lfloor \frac{n-1}{C} \right\rfloor} \cdot \sum_{i=1}^{\left\lfloor \frac{n-1}{C} \right\rfloor} \hat{\theta}_i^{(2)} = \frac{1}{n_C} \sum_{i=1}^{n_C} \hat{\theta}_i^{(2)}$.

$n_C := \left\lfloor \frac{n-1}{C} \right\rfloor$

$E\hat{\theta}_2 = \theta$. Motivation for $\hat{\theta}_2$ is $Var(\hat{\theta}_2) \approx \frac{1}{n_C^2} \sum_{i=1}^{n_C} Var(\hat{\theta}_i^{(2)})$. (if we ignore correlations)
=> $Var(\hat{\theta}_2)$ could be potentially lower than that of any $\hat{\theta}_i^{(2)} \ \forall i$.
=> $D := \hat{\theta}_1 - \hat{\theta}_2$.

3) When the selection is positive => $\hat{\theta}_1 \searrow$ (was explained during lectures) but $\hat{\theta}_2 \nearrow$ as $\bar{m}_{iC} \nearrow$ for large i. and $\bar{m}_{iC}$ has a weight iC. => $D < 0$

4) Analog. for negative selection => $\hat{\theta}_1 \nearrow$ and $\hat{\theta}_2 \searrow$ (as $\bar{m}_{iC} \nearrow$ for small i, but the coefficient is low) => $D > 0$.

CSE 280A
Assignment 3
List 6
Andrey Bzikadze

Q: "Is it possible to devise a statistic based on long haplotype lengths?"

As I asked on Piazza, I'm not too sure what should the answer look like.

Generally if reads do not capture more than one SNV, then the analysis becomes much more complicated — we have almost no information on dependency of various SNV. Something can be done if the coverage is close to uniform —

mut. \ pos:	i	j
0	25	50
1	75	50

=> at least 25 indiv. have both i and j mutation

But this analysis is very limited. If the coverage is far from the uniform, the intersections will be even less powerful!

CSE 280A
Assignment 3
List 7
Andrey Bzikadze

(3) I used wrapper that was published on Piazza, got rid of selection, and simulated samples from each populations: EAS, EUR, AFR.
The full command is on the next page with comments.

Here I just explain the normalization of all parameters.

- $N_e = N_A = 7310$ // we could use whatever normalization here
- AFR - 1st; EUR - 2nd; EAS - 3rd.
- migration rates:

	1	2	3
1	·	m_{AfEu}	m_{AfAs}
2	·	·	m_{EuAs}
3	·	·	·

and multipl. $4N_e$

- generation length = 25 years.
- all times are normalized by $4Ne \cdot gen\ len$.
- we don't need any 'mode' (in script) as we are not simulating selection.
- growth rates are normalized by $4Ne$ (multipl.)

$$\Rightarrow \alpha_{AS} = r_{AS} \cdot 4 \cdot Ne$$
$$\alpha_{EU} = r_{EU} \cdot 4 \cdot Ne$$

- At zero level we need to calculate initial ~~p~~population sizes for EAS & EUR.

$$N_2 = N_{EUR} = N_{EU0} \cdot e^{\alpha_{EU} \cdot T_1}$$
$$N_3 = N_{EAS} = N_{AS0} \cdot e^{\alpha_{AS} \cdot T_1}$$

where T_1 is normalized T_{EuAs}.

- θ, ρ are set up in script according to some papers and are considerably criptic.

List 8 contains command run.
List 9-11 — AFS for AFR, EUR and EAS.
// -I 3 200 0 0 0
for first sample etc..

1) -N 7310 -- Ne
2) -ms 200 100 -- 100 samples, sample size 200=100*2 due to diploidy

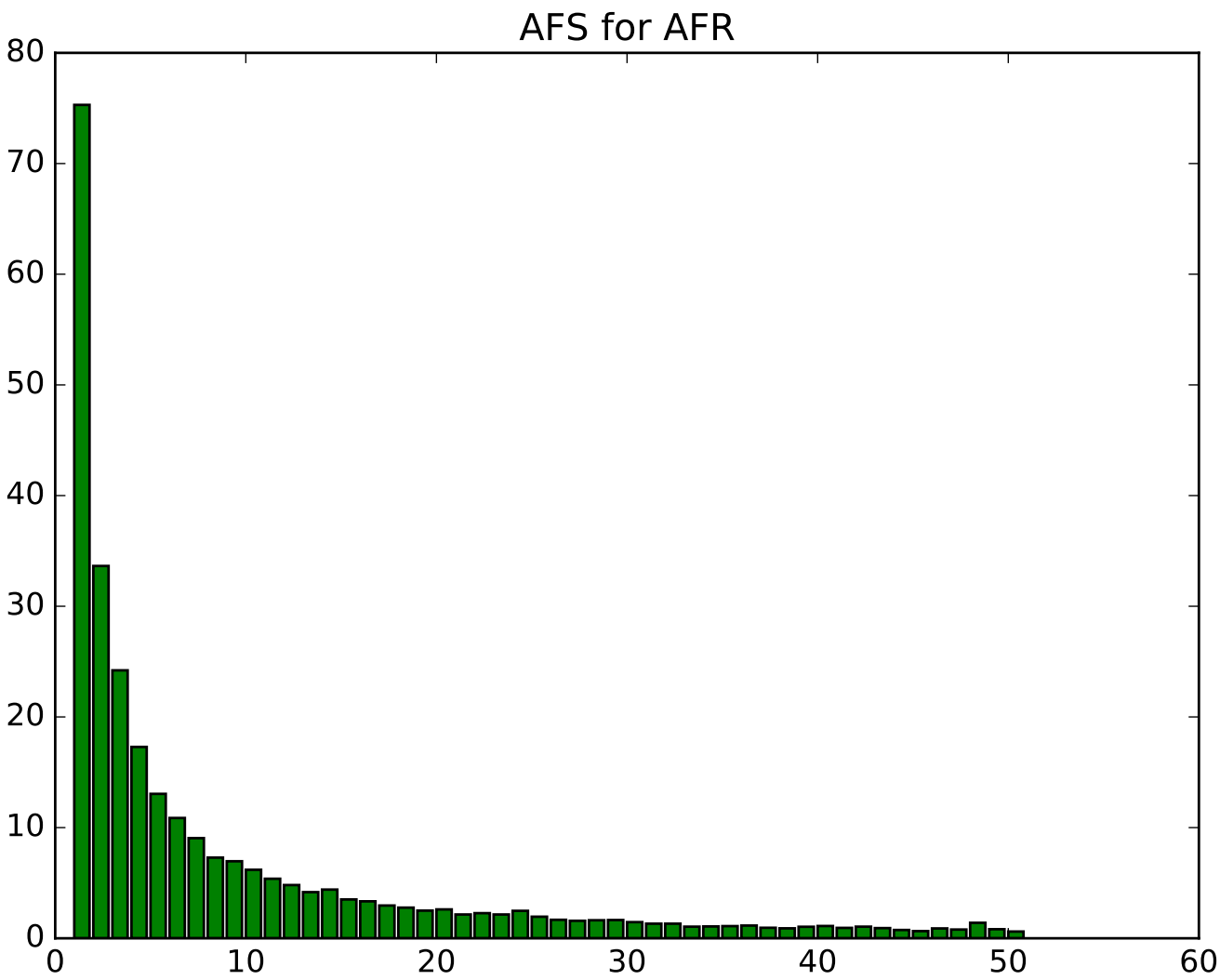
Zero level. Time -- now
3) -I 3 200 0 0 0 -- 3 populations, sample from 1st (other commands for 2,3)
4) -t 36.55 -- theta (acc. to study from script)
5) -r 18.27 -- recombination rate (acc. to study from script)
6) -g 2 111.11 -g 3 140.35 -- growth exponential rates for EUR and EAS
7) -n 1 1.98003 -n 2 4.65646 -n 3 6.27244 -- pop sizes prop to Ne
8) -m 1 2 0.73100 -m 2 1 0.73100 -m 1 3 0.22807 -m 3 1 0.22807 -m 2 3 0.90644 -m 3 2 0.90644 --
setting up migration rates

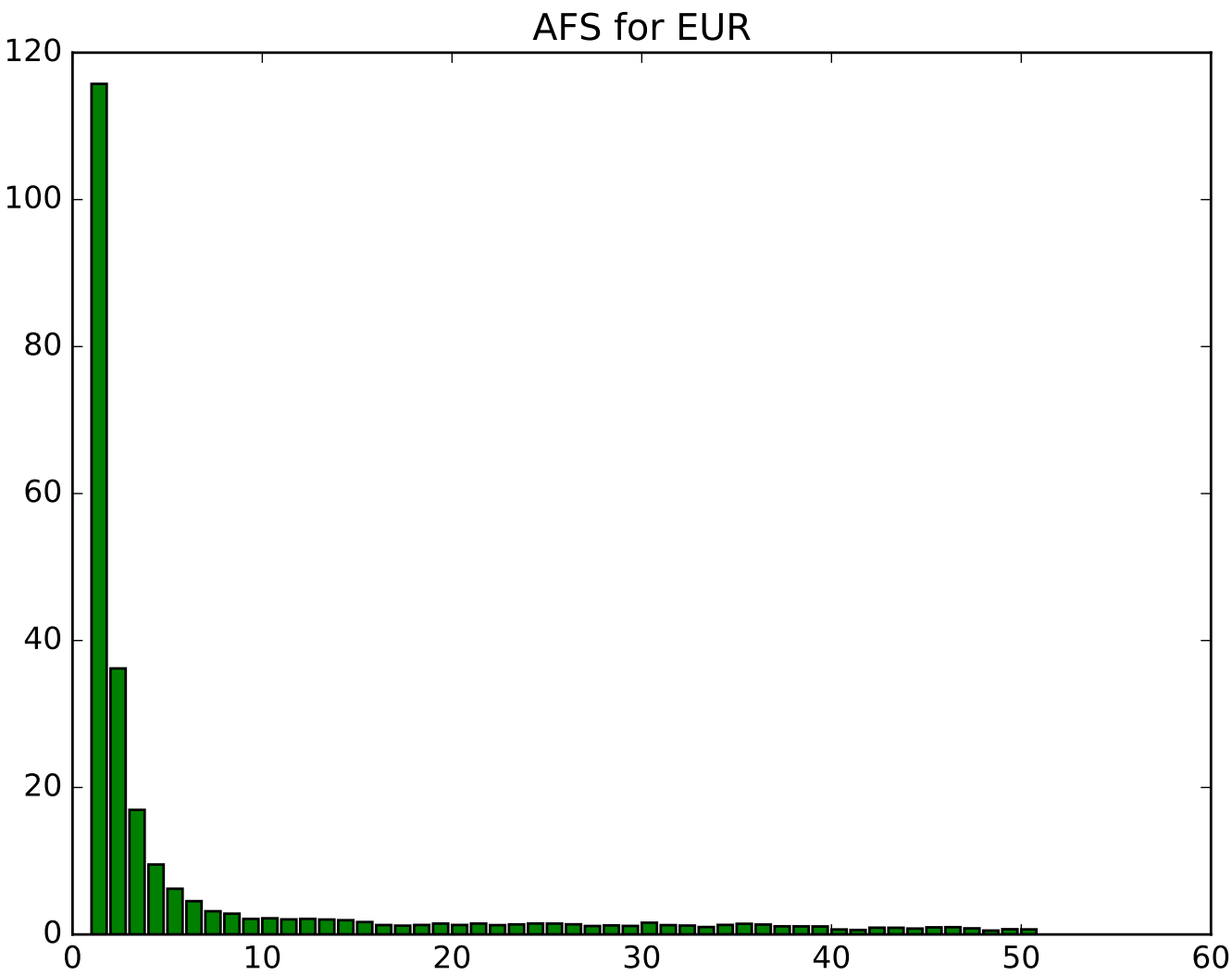
First level. Time -- 23kya
9) -ej 0.03146 3 2 -- Join EAS to EUR subpopulation
10) -en 0.03146 2 0.25458 -- set 1861/Ne as the size of 2nd
11) -em 0.03146 1 2 4.38600 -em 0.03146 2 1 4.38600 -- set new migration rate between AFR & EUR
12)

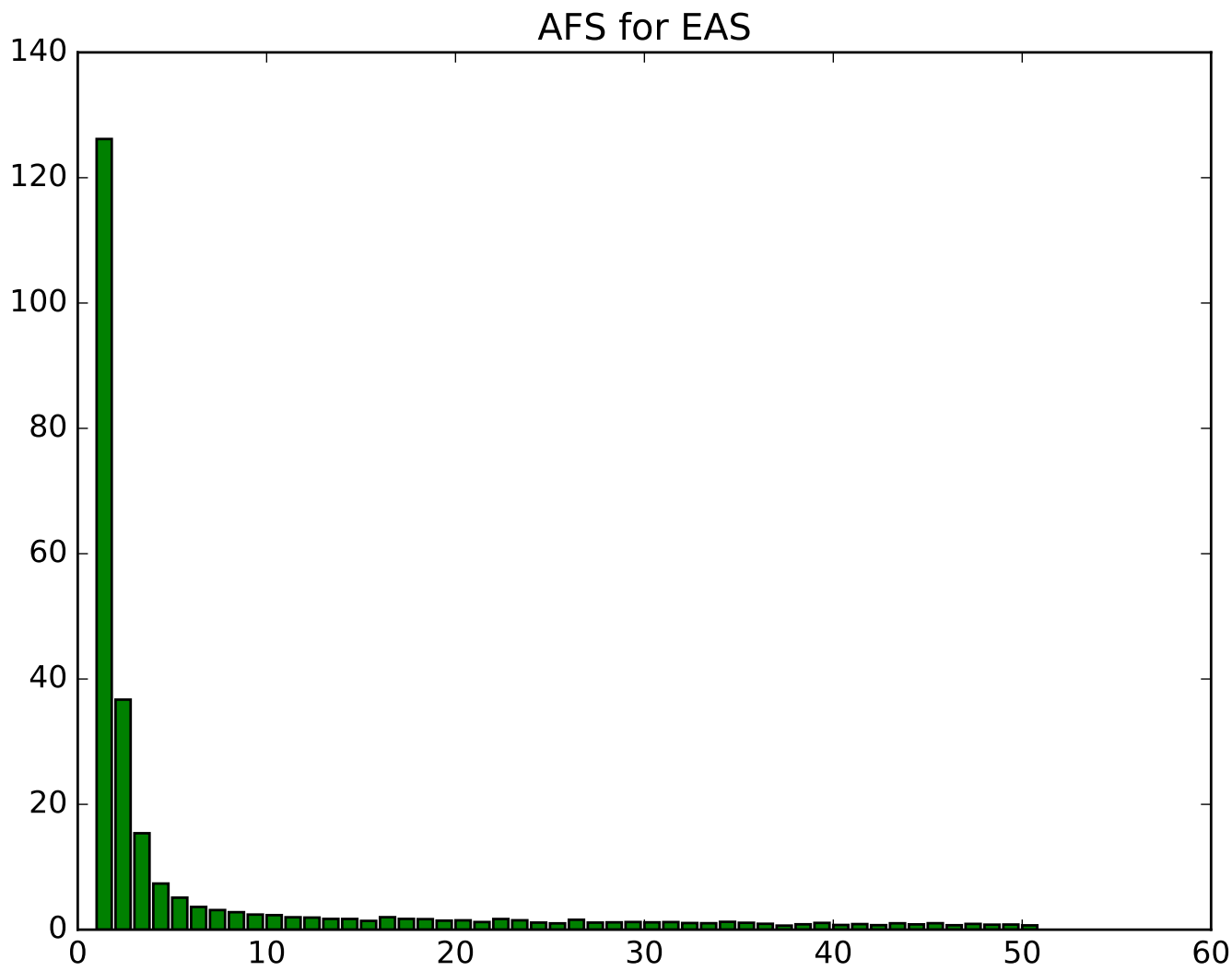
Second level. Time -- 51kya
12) -ej 0.06977 2 1 --- Join 2 to AFR

Third level. Time -- 148kya
13) -en 0.20246 1 1 --- Set at 148kya AFR population to Ne

Full command
java -jar ./msms/lib/msms.jar -N 7310 -ms 200 100 -I 3 200 0 0 0 -t 36.55 -r 18.27 50000 -g 2
111.11 -g 3 140.35 -n 1 1.98003 -n 2 4.65646 -n 3 6.27244 -m 1 2 0.73100 -m 2 1 0.73100 -m 1 3
0.22807 -m 3 1 0.22807 -m 2 3 0.90644 -m 3 2 0.90644 -ej 0.03146 3 2 -en 0.03146 2 0.25458 -em
0.03146 1 2 4.38600 -em 0.03146 2 1 4.38600 -ej 0.06977 2 1 -en 0.20246 1 1 -oAFS onlySummary


AFS for AFR
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60


AFS for EUR
120
100
80
60
40
20
0
0
10
20
30
40
50
60


AFS for EAS
140
120
100
80
60
40
20
0
0
10
20
30
40
50
60

CSE 280A
Assignment 3
List 12
Andrey Bakadze

(4) For this assignment I chose

-I 3 200 200 200.

I'm actually pretty sure that msms gives individuals sorted with respect to population. At the end I explained how I checked it.

For each sample I trained a linear SVM with multiclass rule for optimization. Train/test split is 0.8/0.2.
To test quality I used „accuracy" metric: mean #errors.
It is usually considered a poor metric as is (1) very radical for multiclass classification, (2) does not respect unequal sizes of clusters, (3) cannot be modified easily to support probabilistic classifiers. However, in this assignments (1) the classification is easy, so metric works, (2) populations are equal in size; (3) SVM is not a probabilistic estimator.

For random guess accuracy = $2/3$ (for 3 clusters).
I checked hypothesis that my accuracies are generally lower than $2/3$. // $H_0: E\xi = a$ $t = \sqrt{n}\frac{\bar{x}-a}{s} \xrightarrow[n\to\infty]{} N(0;1)$ //

T test statistic was ≈ -141.4, pvalue $< 10^{-100}$

Why I believe that individuals are sorted by population?
It is highly unlikely that I would get such a good pvalue if the labeling was wrong. To clear all doubts I repeated the analysis with presorting each sample $\Rightarrow$ the labeling is really random.

T test statistic ≈ -0.5339, pvalue ≈ 0.595.

Improvements. In this scenario the classification is very easy and probably shouldn't be improved, but for harder tasks:
(1) Metric: accuracy → Multiclass AUCROC
(2) Classifier: More advanced classifiers. F.e., more flexible kernels
(3) Transformation of data. F.e. Binary → Real valued; Feature-selection.