

① I used msms simulator for this task.

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

In msms I need parameter $-\theta$ 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 $-\theta$ AFS only summary) // trimmed first 25 pos.

List 3 shows the same but with normalized on ~~total sum of~~ total # of mutations.

$$E m_i = \frac{\theta}{i}$$

List 4 shows estimate of θ with the formula $\hat{\theta}_i = \frac{m_i \cdot i}{\sum m_i}$. For $N=\text{const}$ $\hat{\theta}_i$ is an unbiased estimate ($E \hat{\theta}_i = \theta$) for any i . The variance rises with i .

just because ~~the~~ $m_i \downarrow$ with $i \rightarrow \infty$. For N -exp. growing

$\hat{\theta}_i$ is clearly biased and underestimating the real θ .

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

$$\Rightarrow E T_k = \frac{N}{\binom{k}{2}} \Rightarrow E T_k \downarrow \text{as } k \text{ increases.}$$

\Rightarrow more mutations are introduced in early edges (as is prop. to the length)

\Rightarrow more mutations are inherited.

(comparing to N -exp., as we will see)

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

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

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

$\Rightarrow E T_k \uparrow$ with k increasing. \Rightarrow

\Rightarrow more mutations are introduced in later edges \Rightarrow

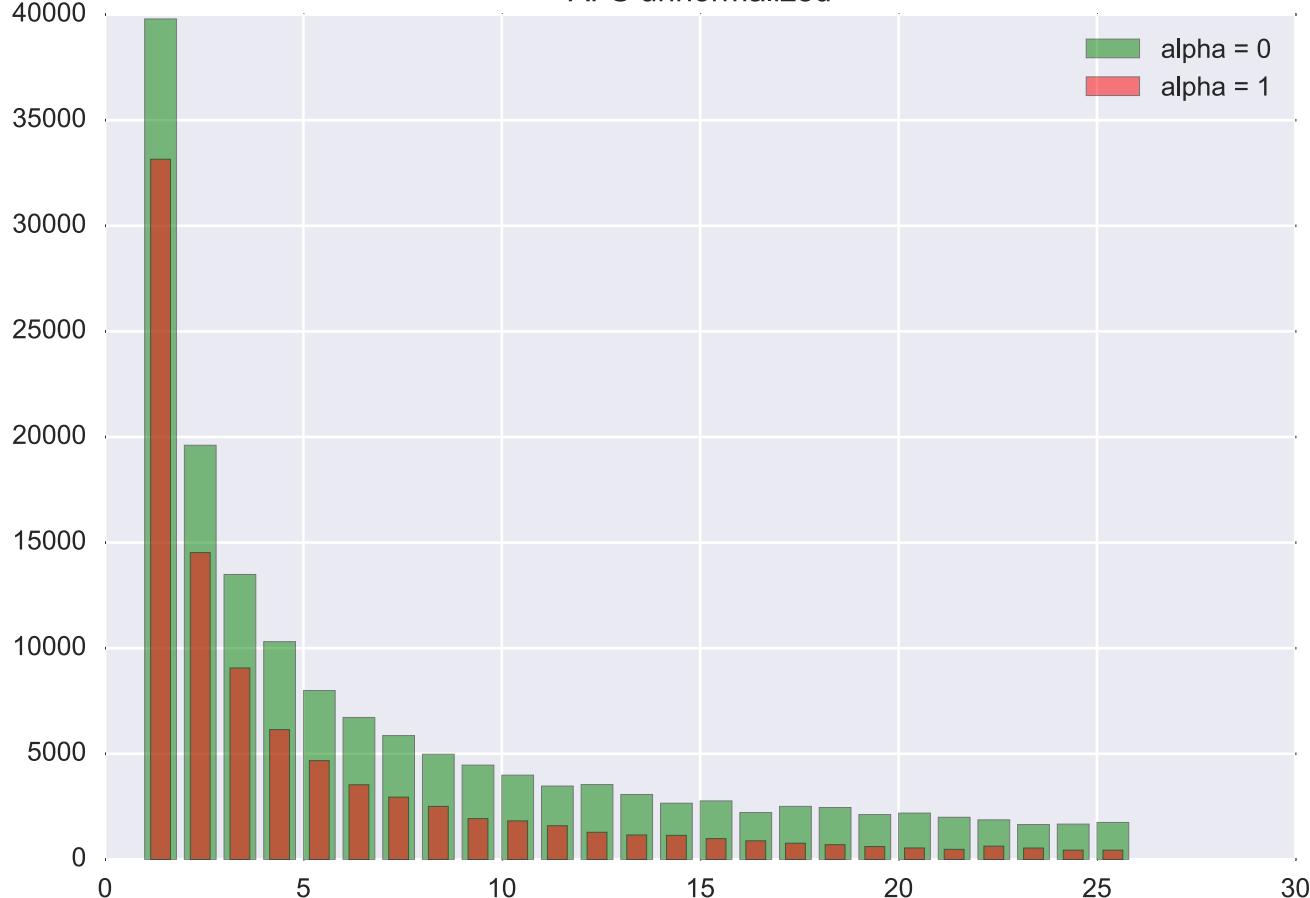
\Rightarrow more less (compared to $N=\text{const}$) mutations are inherited.

CSE 280A
Assignment 3

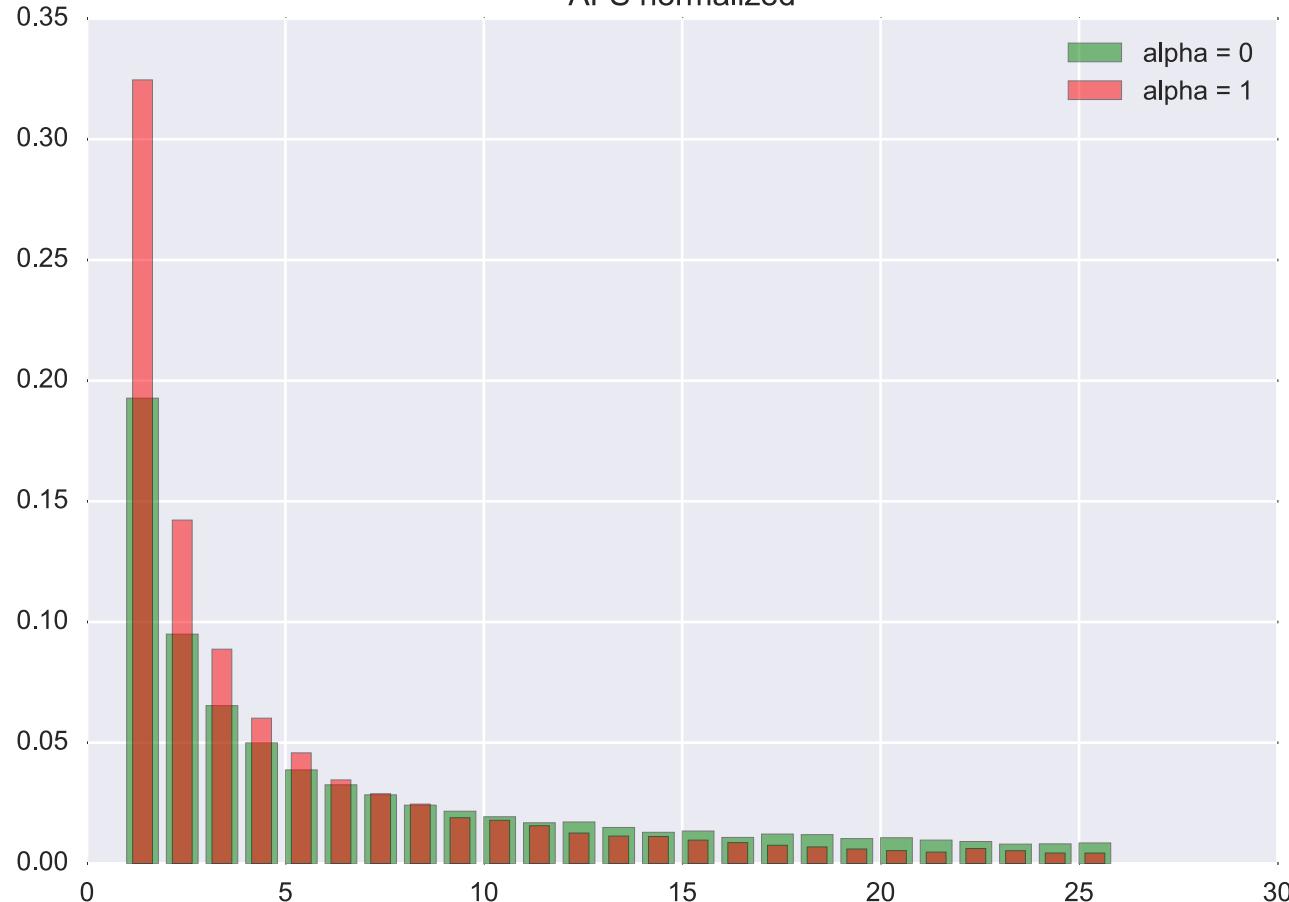
List 1

Andrey Bzikadze

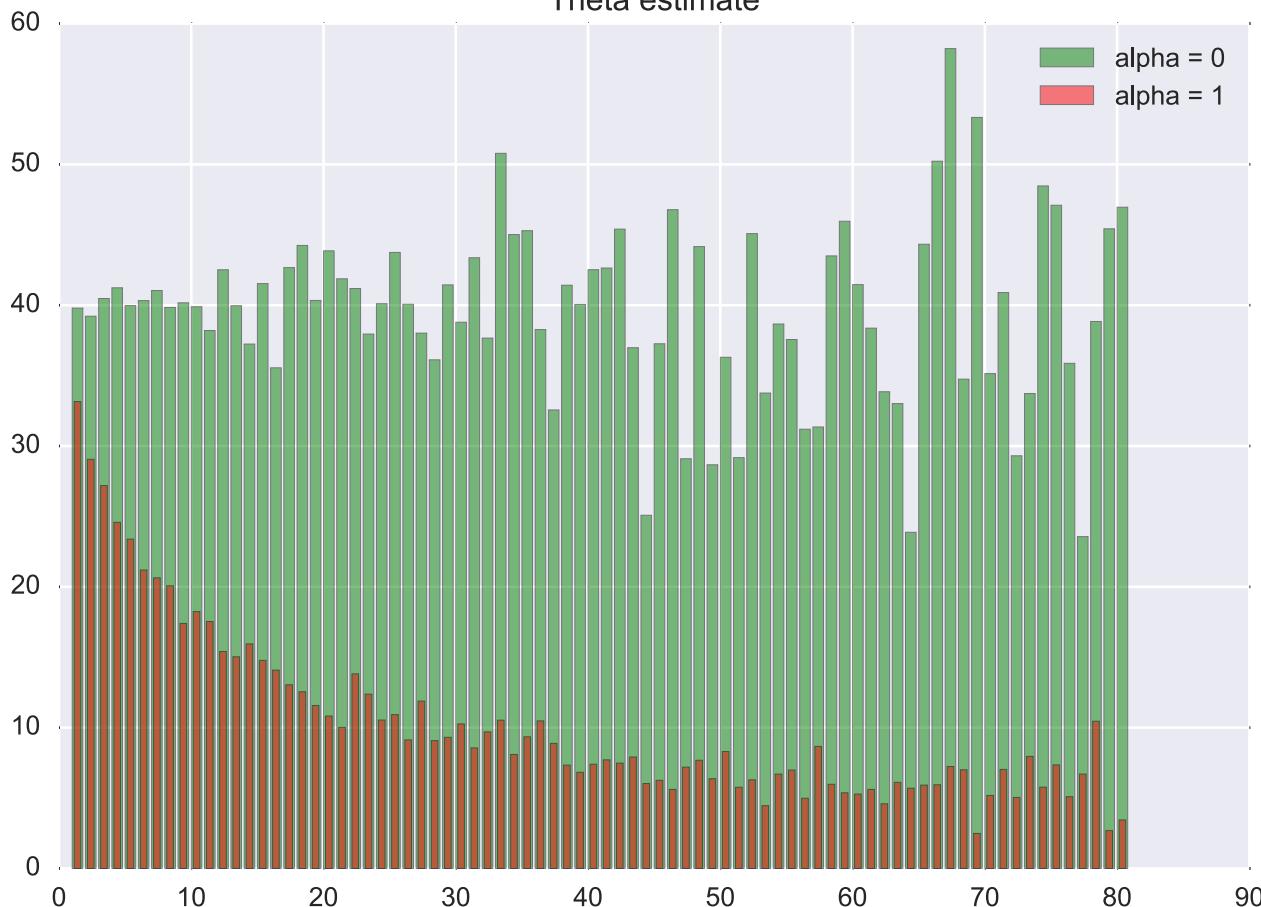
AFS unnormalized



AFS normalized



Theta estimate



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 $Z = \frac{D}{\text{std}(D)} \xrightarrow{n \rightarrow \infty} N(0; 1)$ under null hypothesis that there is no selection, $\Rightarrow \text{pvalue} = 2 \min(\Phi'(z), 1 - \Phi'(z))$.

1) I propose $\hat{\theta}_1 - \text{a Tajima's } k$. From assignment 2.5

we know that it looks like $k = \sum_{j=1}^m f_j (1-f_j)$, where f_j is the frequency of ^{mutation} allele at site j . I'm not sure that $E_k = \theta$ (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. First, suppose that $n_i = h_i \forall i$, and reads are error-free.

We as usual denote m_i as # mutations with exactly i (one) reads $\Rightarrow m_i$ (from data) : $\bar{m}_i = 0 \quad \forall i < C$ 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 $\bar{m}_c > \bar{m}_i \quad \forall i < C$ and AFS could look like this
(under no selection)

The same when $n_i \neq h_i$ exactly, but close,

\Rightarrow We can define estimates: $\hat{\theta}_i^{(2)} = i \cdot C \cdot \bar{m}_{ic}$

It is known that $E \hat{\theta}_i^{(2)} = \theta_2$ // $iC \leq n-1 \Rightarrow i \leq \frac{n-1}{C}$

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

$E \hat{\theta}_2 = \theta$. Motivation for $\hat{\theta}_2$ is $\text{Var}(\hat{\theta}_2) \approx \frac{1}{n_c^2} \sum_{i=1}^{n_c} \text{Var}(\hat{\theta}_i^{(2)})$.
(if we ignore correlations)

$\Rightarrow \text{Var}(\hat{\theta}_2)$ could be potentially lower than that of any $\hat{\theta}_i^{(2)}$, $\forall i$.

$\Rightarrow D := \hat{\theta}_1 - \hat{\theta}_2$.

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

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

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 SNVs. Something can be done if the coverage is close to uniform —

		pos:	i	j
		mut.		
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!

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 ~~will~~ 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_{EUSA}	
3			.

and multipl. $4N_e$

- generation length = 25 years.
- all times are normalized by $4N_e \cdot \text{gen len.}$
- we don't need any "mode" (in script) as we are not simulating selection.
- growth rates are normalized by $4N_e$ (multipl.)
 $\Rightarrow d_{AS} = r_{AS} \cdot 4N_e$
 $d_{EU} = r_{EU} \cdot 4N_e$
- At zero level we need to calculate initial 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 cryptic.

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

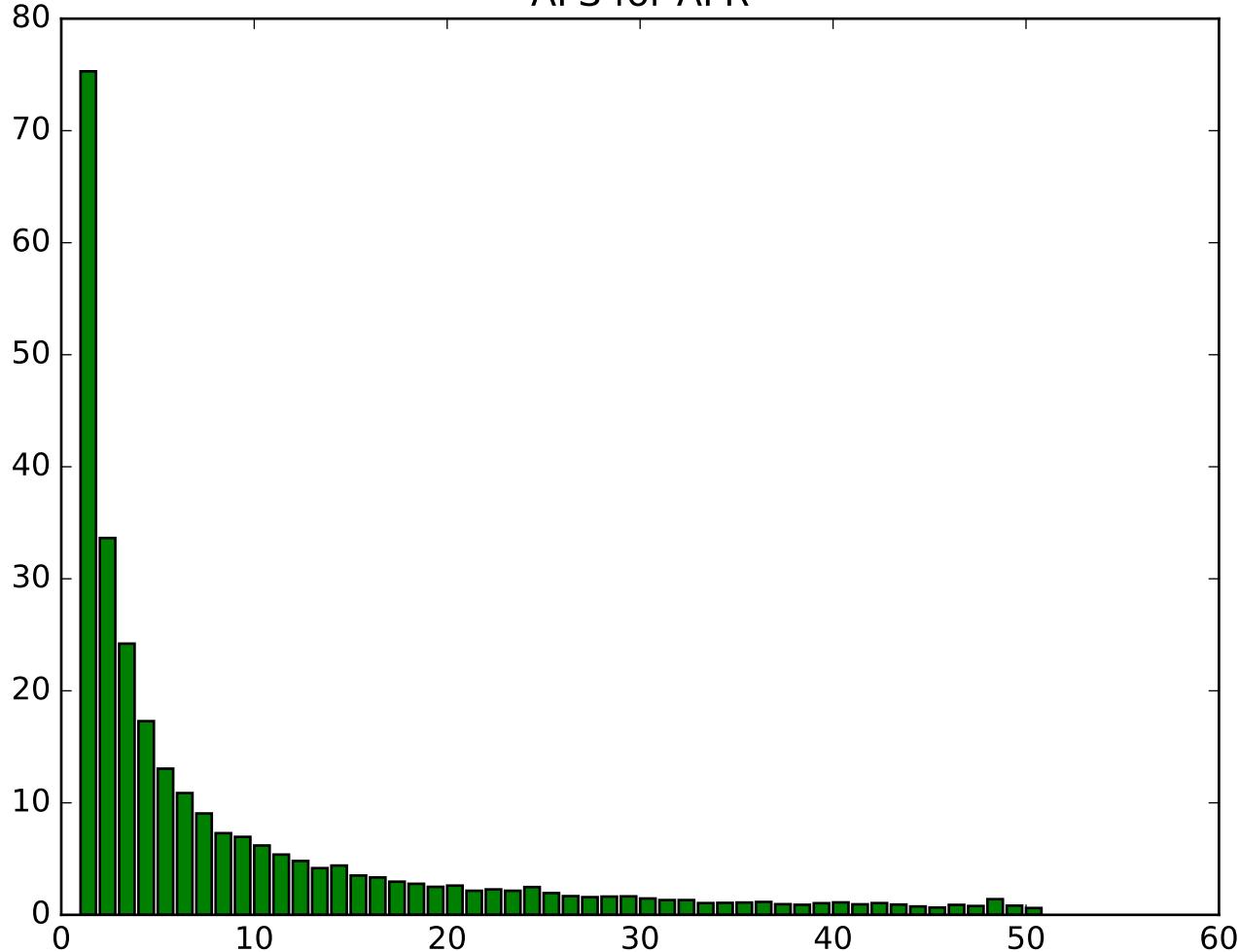
Third level. Time -- 148kya

13) -en 0.20246 1 1 --- Set at 148kya AFR population to Ne

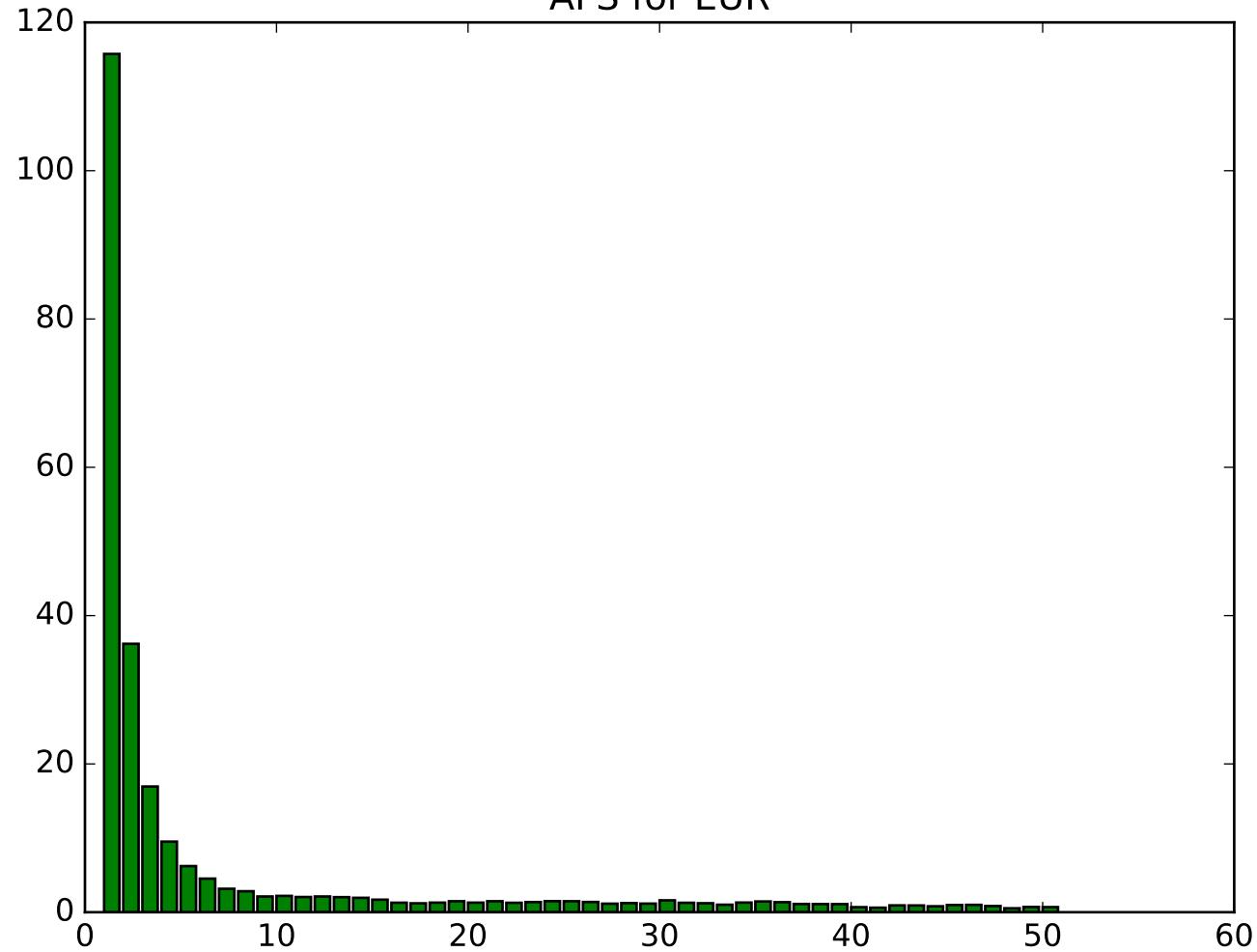
Full command

```
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
```

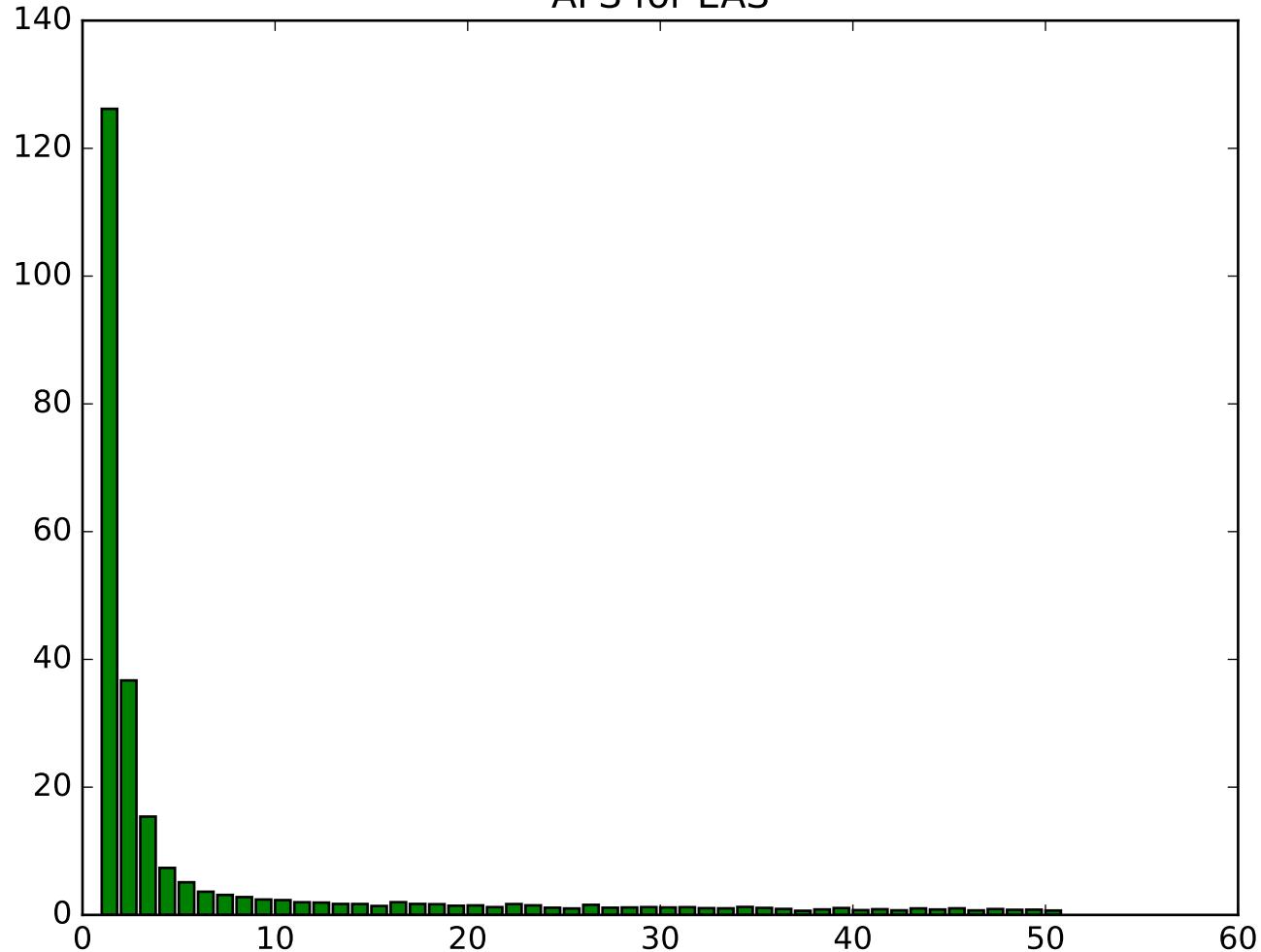
AFS for AFR



AFS for EUR



AFS for EAS



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 (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 assignment (1) the ^{classification} solution is easy, so metric works, (2) populations are equal in size; (3) SVM is not a probabilistic estimator.

For random guess accuracy = $\frac{2}{3}$ (for 3 classes).

I checked hypothesis that my accuracies are generally lower than $\frac{2}{3}$. // $H_0 : E\gamma = \frac{2}{3} \quad t = \sqrt{n} \frac{\bar{x} - \frac{2}{3}}{\sigma} \xrightarrow{n \rightarrow \infty} N(0; 1)$ //

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 permuting 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 \rightarrow Multiclass AUCROC

(2) Classifier: More advanced classifiers. F.e., more flexible kernels

(3) Transformation of data. F.e. Binary \rightarrow Real valued; Feature-selection.