Scaling Genotype-based Genetic Variation Discovery to Millions of Genomes

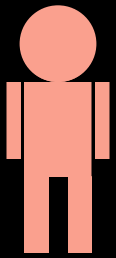
Ryan M. Layer, Aaron R. Quinlan

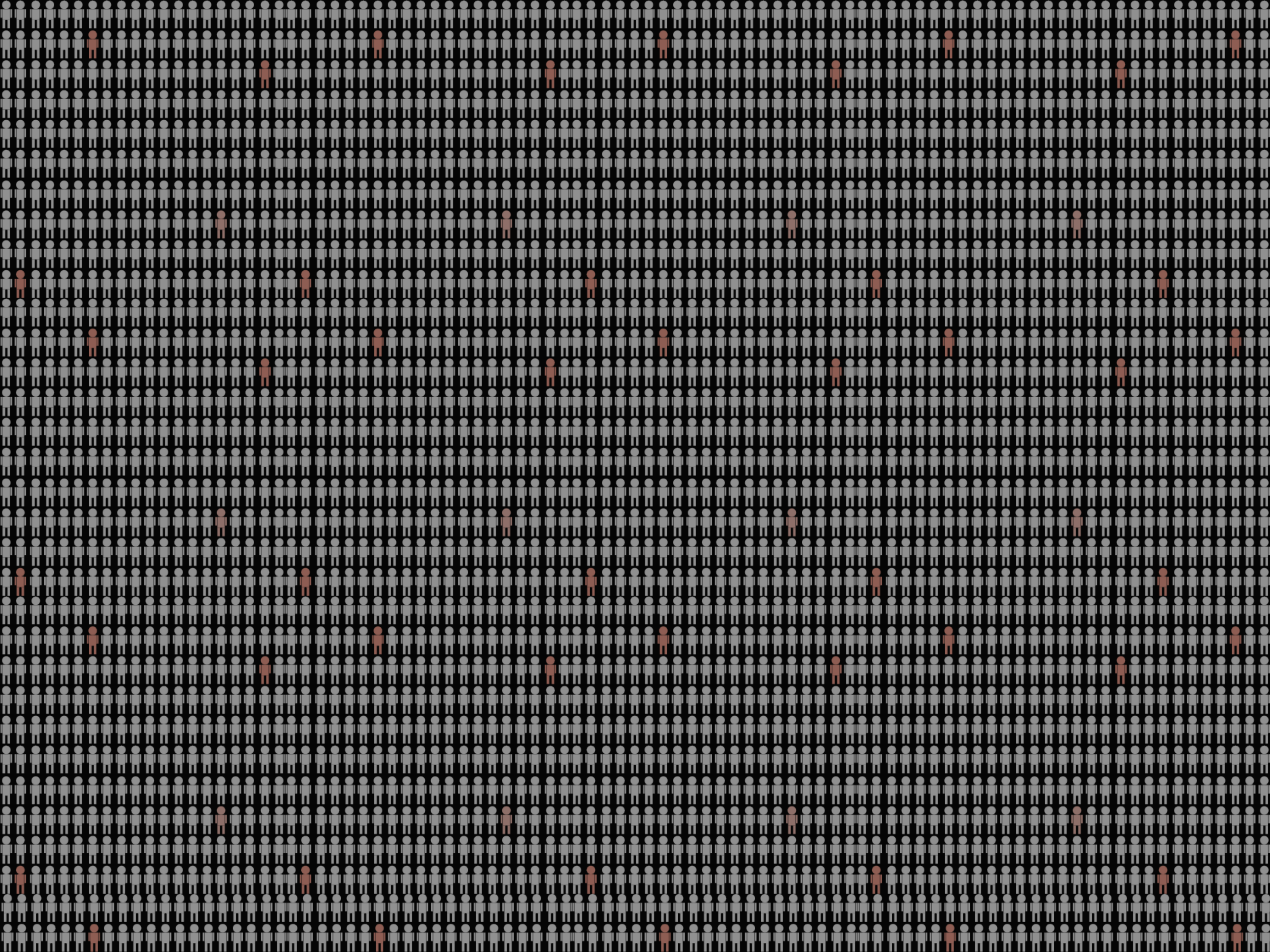
University of Virginia

rl6sf@virginia.edu

@ryanlayer

https://github.com/ryanlayer/gqt





Variant Call Format (VCF)

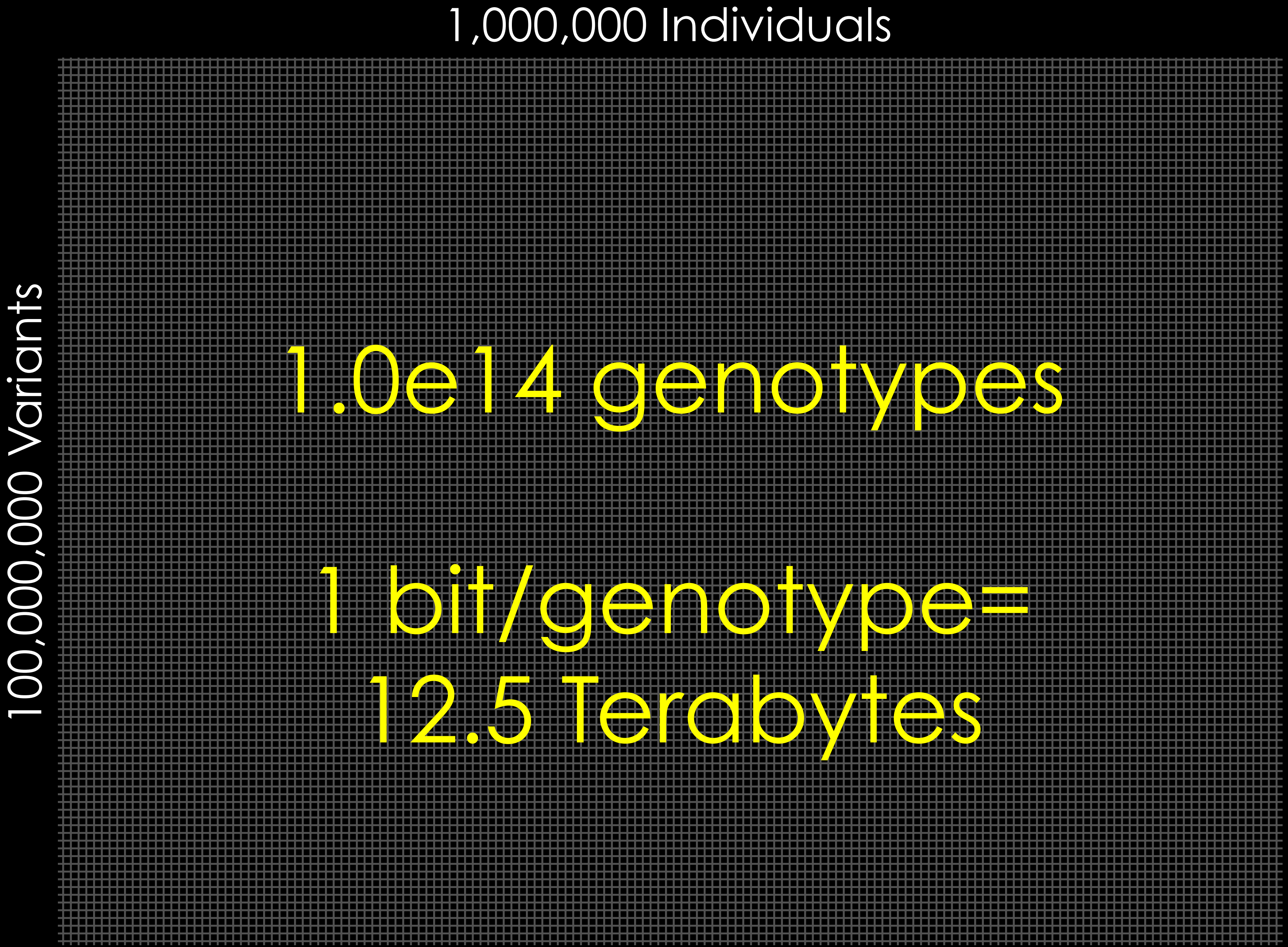
Variant Location/ID Individual Genotypes

```
20   62553    rs114190700    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   62588    rs184741218    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   62731    rs34147676     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0
20   62783    rs189195684    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   62821    rs180933038    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   62880    rs199513831    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   62946    rs183567118    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63008    rs147934693    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63054    rs116457849    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63231    rs6076506      0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 1|0
20   63233    rs141722618    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63244    rs6139074      0|0 0|0 0|0 0|1 0|0 0|1 0|1 0|0 0|0 0|1 0|0 0|1 0|0 0|0 0|0 0|0 0|1
20   63310    rs189736466    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63351    rs181305519    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63360    rs186156309    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63426    rs147063585    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63452    rs115017123    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63521    rs191905748    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63541    rs117322527    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63559    rs138359120    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63696    rs149160003    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63729    rs181483669    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63733    rs75670495     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63799    rs1418258      0|0 0|0 0|0 0|1 0|0 0|1 0|1 0|1 0|0 0|1 0|0 0|1 0|0 0|0 0|1 0|0 0|1
20   63808    rs76004960     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   63967    rs116770801    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   64016    rs143263863    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20   64062    rs148297240    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
```

Variant Call Format (VCF)

Variant Location/ID Individual Genotypes

```
20    62553    rs114190700    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    62588    rs184741218    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    62731    rs34147676     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0
20    62783    rs189195684    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    62821    rs180933038    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    62880    rs199513831    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    62946    rs183567118    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63008    rs147934693    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63054    rs116457849    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63231    rs6076506      0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 1|0
20    63233    rs141722618    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63244    rs6139074      0|0 0|0 0|0 0|1 0|0 0|1 0|1 0|0 0|0 0|1 0|0 0|1 0|0 0|0 0|0 0|0 0|1
20    63310    rs189736466    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63351    rs181305519    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63360    rs186156309    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63426    rs147063585    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63452    rs115017123    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63521    rs191905748    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63541    rs117322527    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63559    rs138359120    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63696    rs149160003    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63729    rs181483669    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63733    rs75670495     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63799    rs1418258      0|0 0|0 0|0 0|1 0|0 0|1 0|1 0|1 0|0 0|1 0|0 0|1 0|0 0|0 0|1 0|0 0|1
20    63808    rs76004960     0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    63967    rs116770801    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    64016    rs143263863    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
20    64062    rs148297240    0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0
```



1,000,000 Individuals

100,000,000 Variants

1.0e14 genotypes

1 bit/genotype=
12.5 Terabytes

Binary Encoding

char = 1 byte (8 bits)

↓

```
20 62553 rs114190700 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0
```

16 genotypes = 16*3*8 = 384 bits

24 bit/genotype = 300 Terabytes

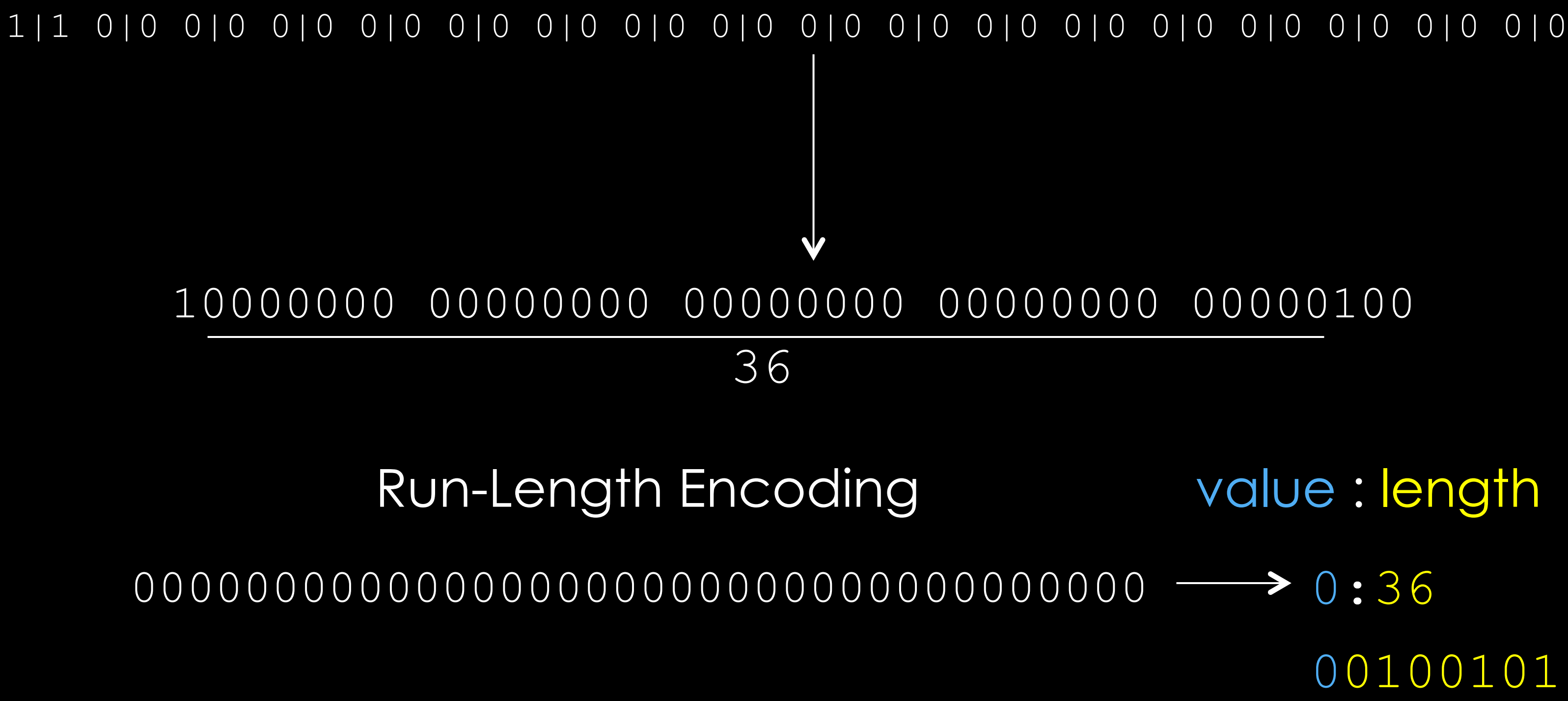
Genotype	VCF	Binary
Homozygous Reference	0|0	00
Heterozygous	0|1	01
Homozygous Alternate	1|1	10
Unknown		11

```
00000000 00000001 00000000 00000100
```

16 genotypes = 16*2 = 32bits

2 bit/genotype = 25 Terabytes

Data Compression



1|1 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0

10000000 00000000 00000000 00000000 00000100

36

Run-Length Encoding

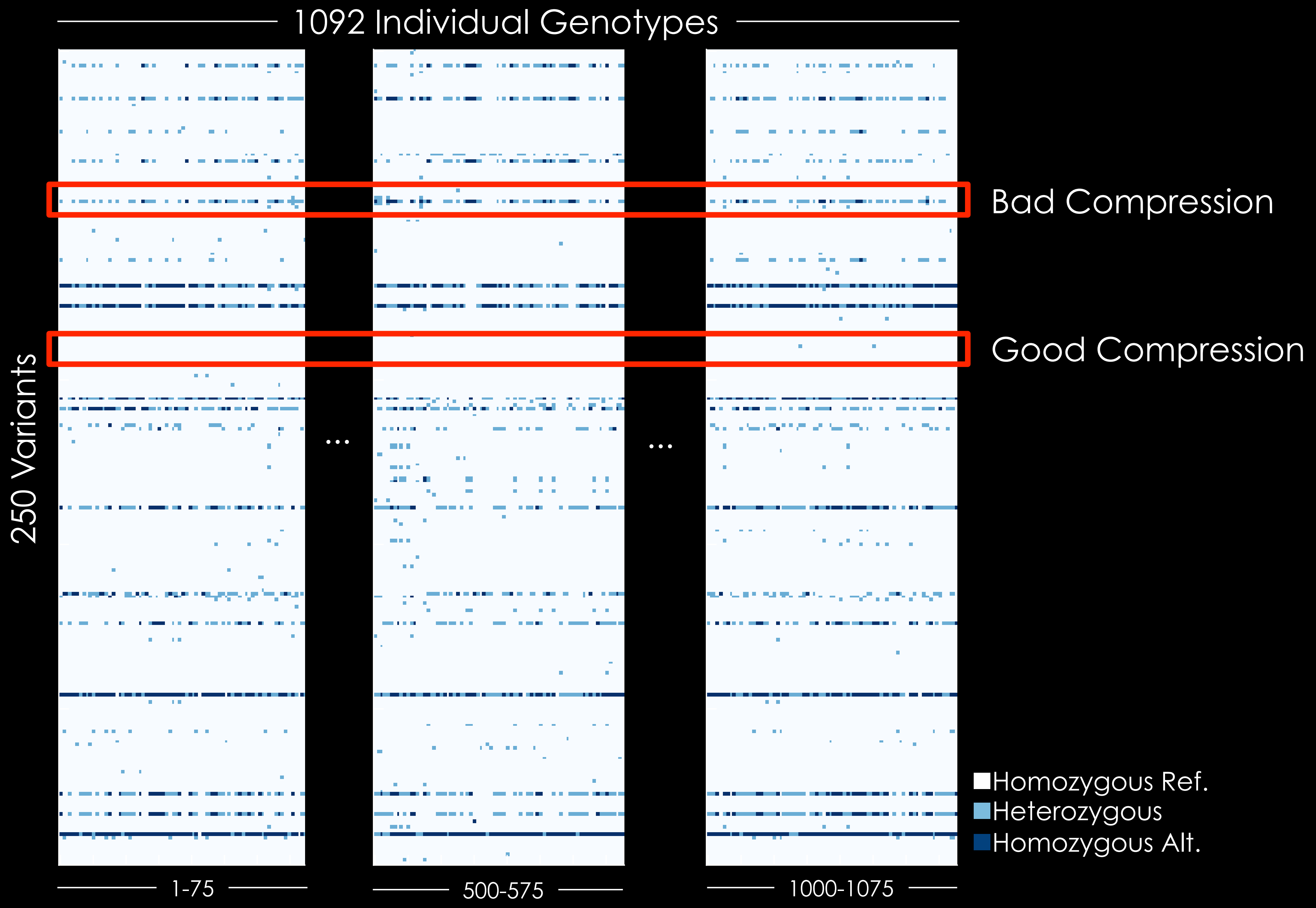
value : length

000000000000000000000000000000000000

0:36

00100101

1000 Genomes chr20

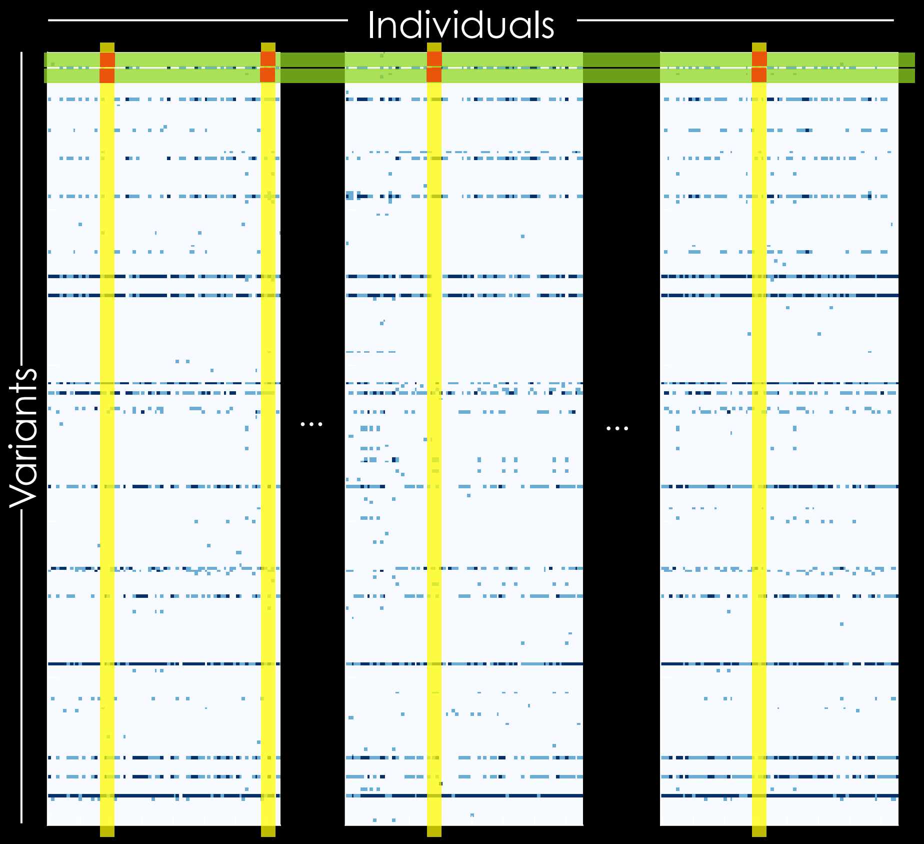


Bad Compression

Good Compression

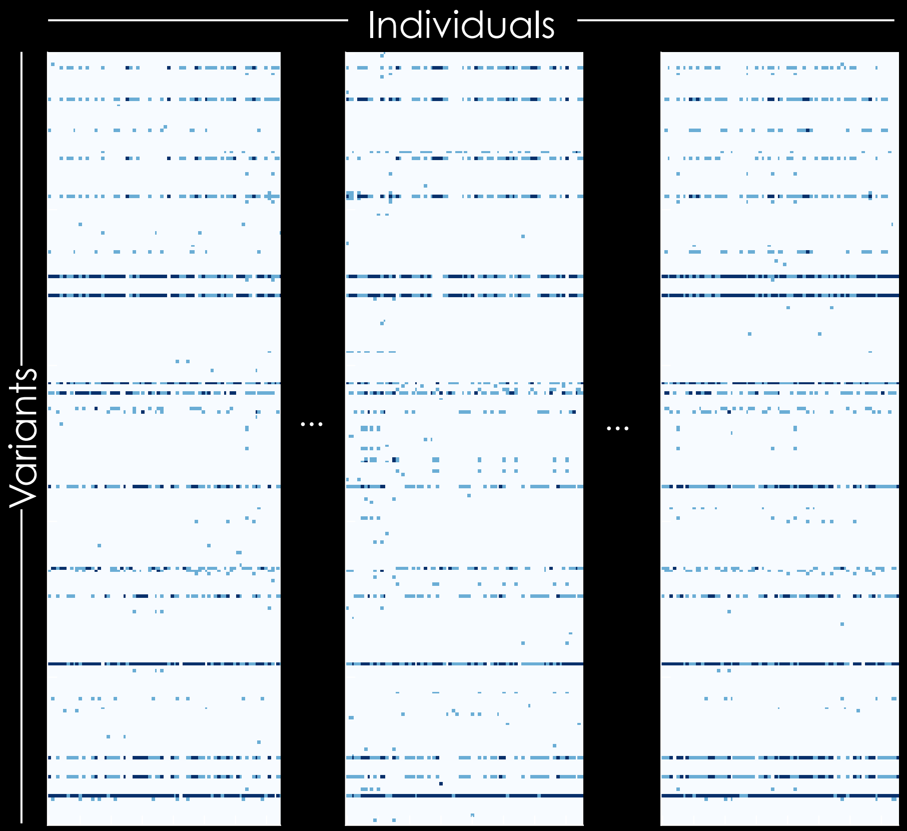
Query: Find the variants that are heterozygous in all affected individuals

```
for each variant in variants
    genotypes = readline()
    hit = true
    for each individual in affected_individuals
        if genotypes[individual] != homozygous_ref
            hit = false
    if hit == true
        print variant
```



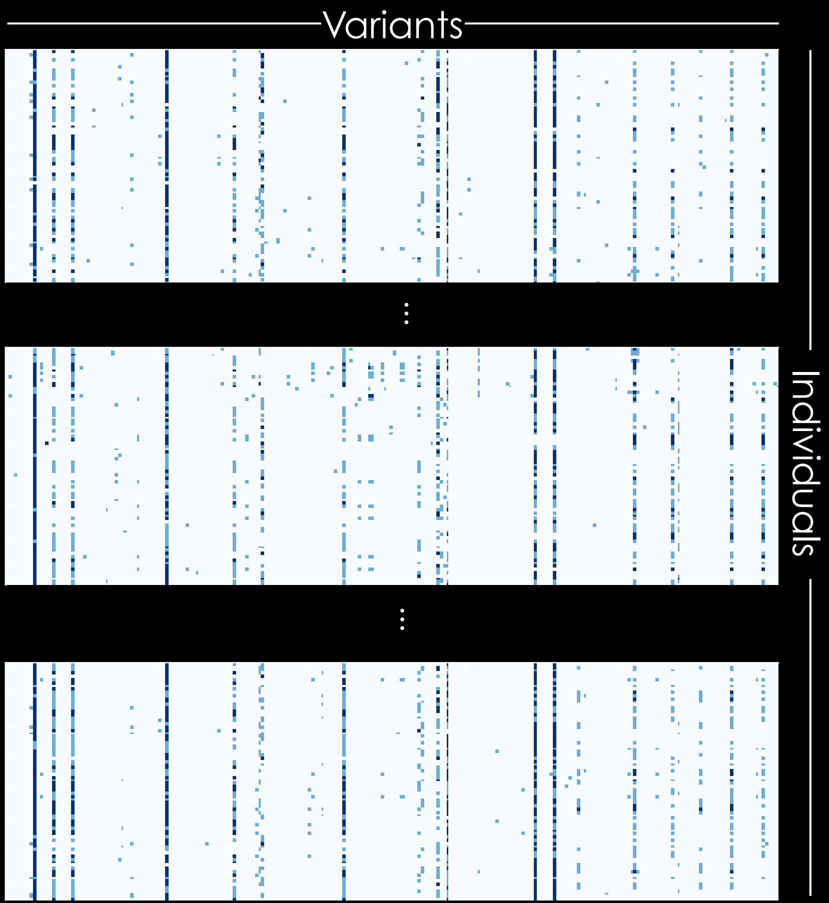
Query: Find the variants that are heterozygous in all affected individuals

```
for each variant in variants
    genotypes = readline()
    hit = true
    for each individual in affected_individuals
        if genotypes[individual] != homozygous_ref
            hit = false
    if hit == true
        print variant
```



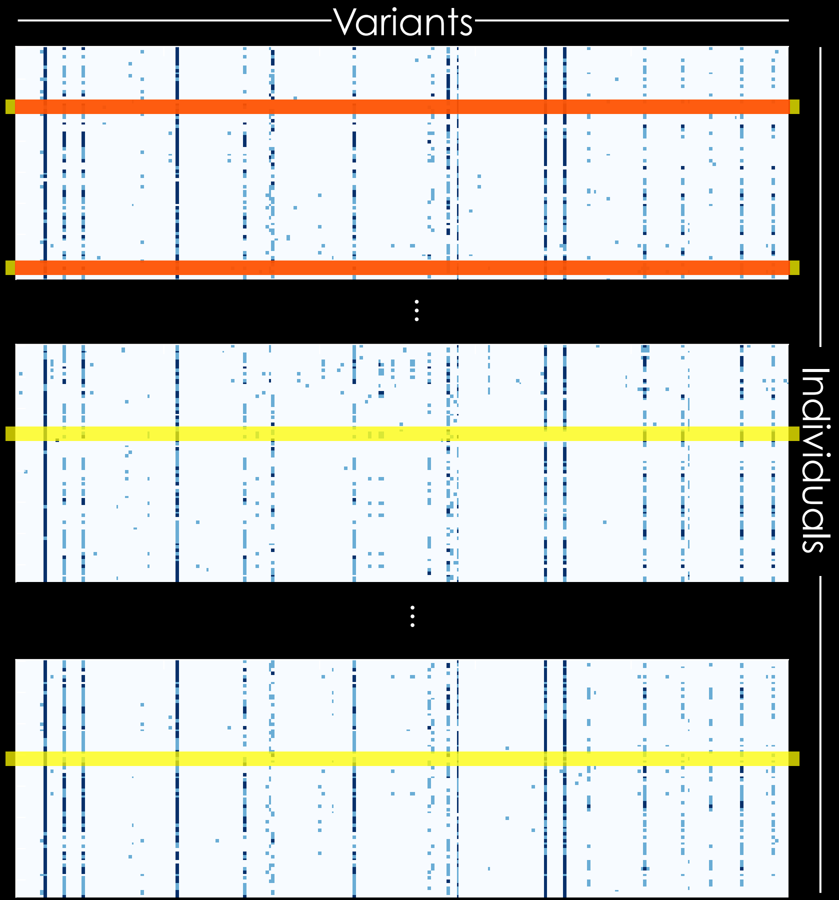
Query: Find the variants that are heterozygous in all affected individuals

```
hits = [true, true, ..., true]
for each individual in affected_individuals
    genotypes = readline()
    for each variant in variants
        if genotypes[variants] != homozygous_ref
            hit[variant] = false
for each hit in hits:
    if hit == true print variant
```

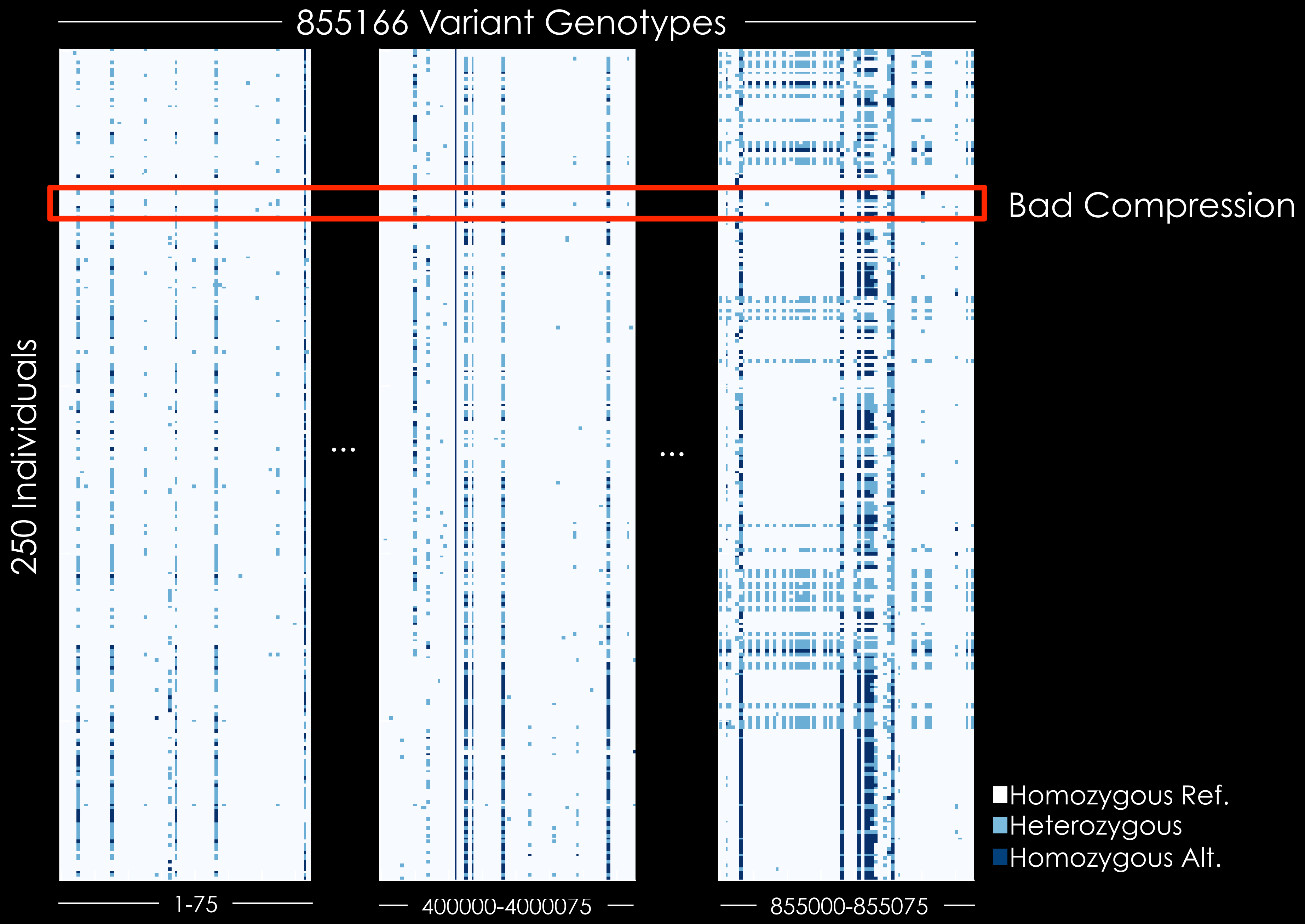


Query: Find the variants that are heterozygous in all affected individuals

```
hits = [true, true, ..., true]
for each individual in affected_individuals
     genotypes = readline()
     for each variant in variants
          if genotypes[variants] != homozygous_ref
               hit[variant] = false
for each hit in hits:
     if hit == true print variant
```



1000 Genomes chr20

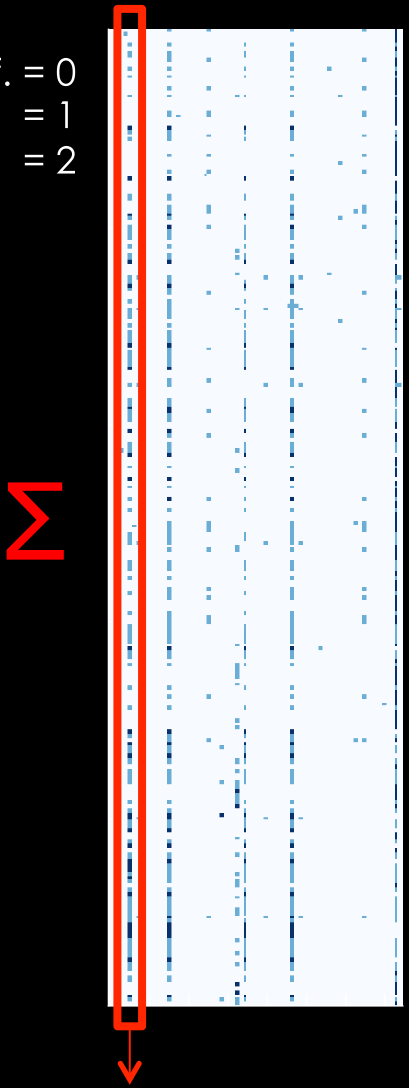


Sort Variants by Allele Frequency

⬜ Homozygous Ref. = 0

🟦 Heterozygous = 1

■ Homozygous Alt. = 2



Σ

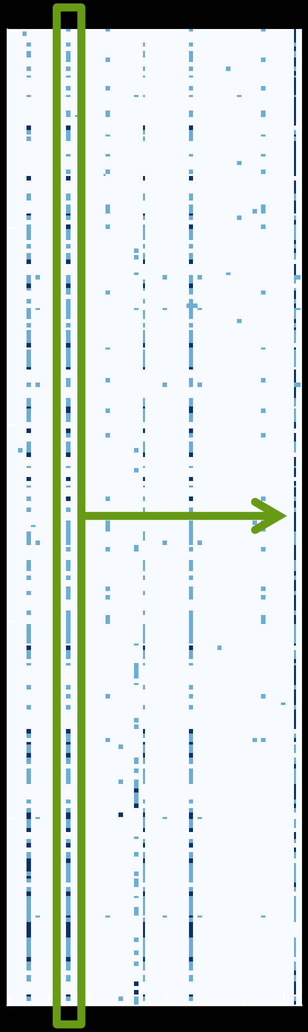
Allele Frequency

Sort Variants by Allele Frequency

□ Homozygous Ref. = 0

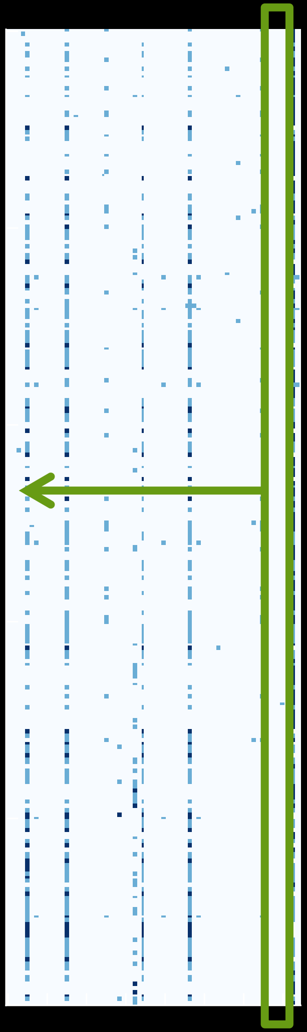
▨ Heterozygous = 1

■ Homozygous Alt. = 2



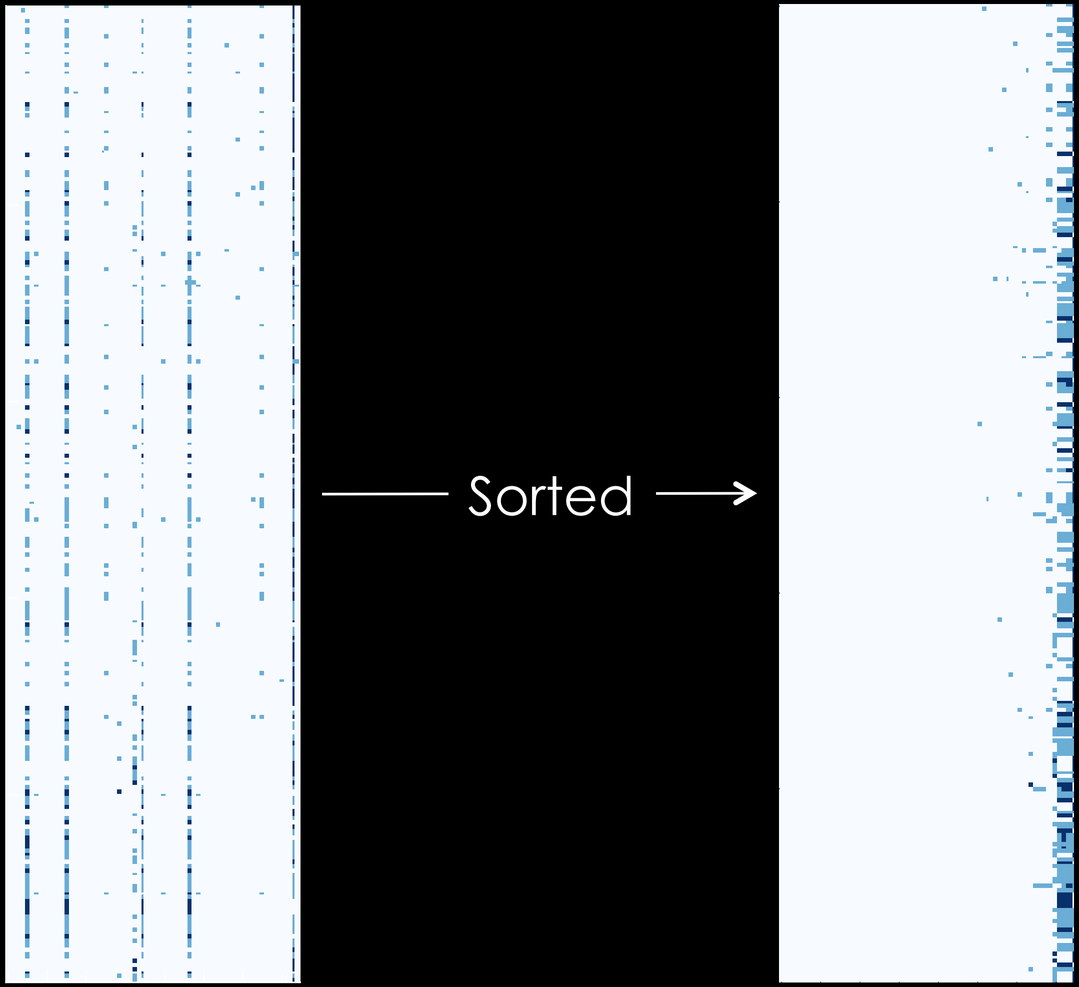
Sort Variants by Allele Frequency

⬜ Homozygous Ref. = 0
🟦 Heterozygous = 1
■ Homozygous Alt. = 2



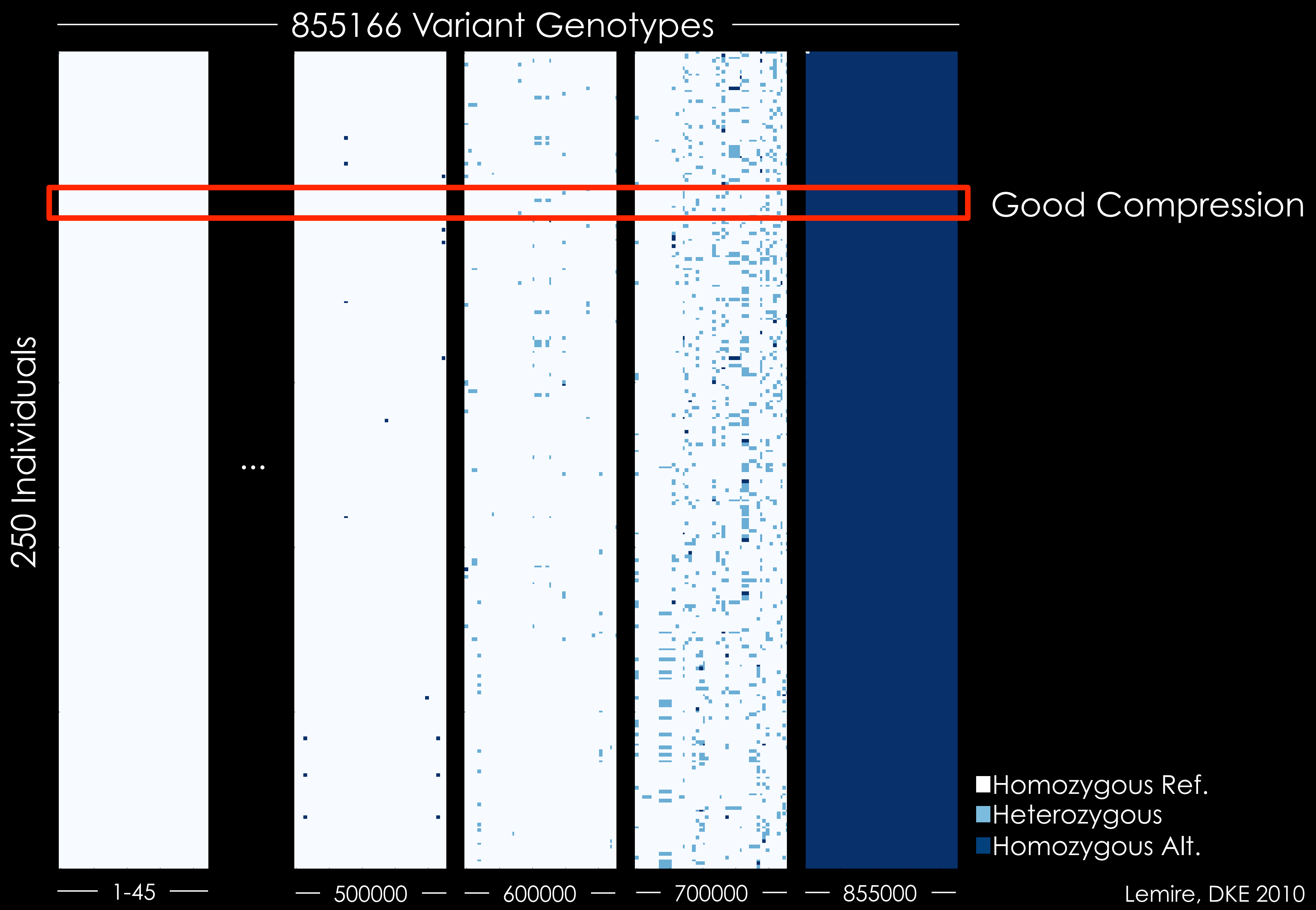
Sort Variants by Allele Frequency

⬜ Homozygous Ref. = 0
🟦 Heterozygous = 1
■ Homozygous Alt. = 2



Sorted →

Allele Frequency Sorted 1000 Genomes chr20



Lemire, DKE 2010

Query: Find the variants that are heterozygous in individuals 1, 2, and 3

```
1   1|1 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0 0|1 0|0 0|0 1|1 0|0
2   0|0 0|0 0|0 1|1 0|1 0|0 0|0 0|0 0|1 0|0 0|0 0|1 0|0 0|0 1|1 0|0
3   0|1 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|1 0|0 0|0 0|0 0|0 0|0 0|0 0|0
-------------------------------------------------------------------
R    F   F   F   F   F   F   F   F   T   F   F   F   F   F   F   F
```

```
R = [T, T, ..., T]
for each individual in [1,2, 3]
    goto(individual)
    genotype = readline()
    for i = 1...|genotypes|
        if genotype[i] != HETEROZYGOUS then R[i] = F
```

```
1   10000000000000000100000100001000
2   00000010010000000100000100001000
3   01000000000001000100000000001000
```

```
R = [T, T, ..., T]
for each individual in [1,2, 3]
    goto(individual)
    genotype = readline()
    for i = 1...|genotypes|
        if (genotype >> (31 - i)) & 1 != HETEROZYGOUS then R[i] = F
```

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

```
1    1|1 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0 0|0 0|1 0|0 0|0 1|1 0|0
```

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

<table>
<tr><td>1</td><td>1|1</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|1</td><td>0|0</td><td>0|0</td><td>0|1</td><td>0|0</td><td>0|0</td><td>1|1</td><td>0|0</td></tr>
<tr><td>Homo Ref</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

1	1|1	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|1	0|0	0|0	0|1	0|0	0|0	1|1	0|0
Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

1	1|1	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|1	0|0	0|0	0|1	0|0	0|0	1|1	0|0
Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

<table>
<tr><th>1</th><th>1|1</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>1|1</th><th>0|0</th></tr>
<tr><td>Homo Ref</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Het</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Homo Alt</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Unknown</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bitmap Index: map records to bit arrays then answer queries using bitwise logical operations

<table>
<tr><th>1</th><th>1|1</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>1|1</th><th>0|0</th></tr>
<tr><td>Homo Ref</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Het</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Homo Alt</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Unknown</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><th>2</th><th>0|0</th><th>0|0</th><th>0|0</th><th>1|1</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>1|1</th><th>0|0</th></tr>
<tr><td>Homo Ref</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Het</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Homo Alt</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Unknown</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><th>3</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|1</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th><th>0|0</th></tr>
<tr><td>Homo Ref</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Het</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Homo Alt</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Unknown</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Query: Find the variants that are heterozygous in individuals 1, 2, and 3

1	Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Homo Ref	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Homo Ref	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1
	Het	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	Homo Alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Query: Find the variants that are heterozygous in individuals 1, 2, and 3

1	Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Homo Ref	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Homo Ref	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1
	Het	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	Homo Alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	AND	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Query: Find the variants that are either heterozygous or homozygous alternate in individuals 1, 2, and 3

1	Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Homo Ref	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Homo Ref	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1
	Het	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	Homo Alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Query: Find the variants that are either heterozygous or homozygous alternate in individuals 1, 2, and 3

	Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
1	Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
	Homo Ref	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1
2	Het	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
	Homo Alt	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
	Homo Ref	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1
3	Het	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	Homo Alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Query: Find the variants that are either heterozygous or homozygous alternate in individuals 1, 2, and 3

	Homo Ref	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
1	Homo Alt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
	Homo Ref	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1
	Het	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
2	Homo Alt	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
	Homo Ref	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1
	Het	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
3	Homo Alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OR	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	AND	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

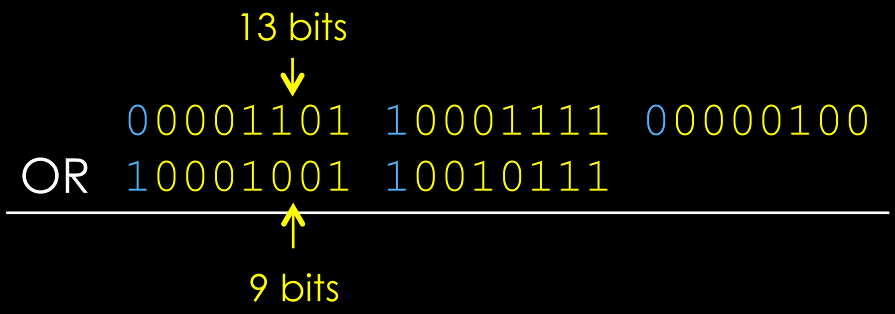
```
VCF       1|1 0|0 0|0 0|0 0|0 0|0 0|0 0|0 0|1 0|0      24 bits/genotype

Binary    10000000 00000000 01000001 00001000          2 bits/genotype

          01111111 01101101
Bit Map   00000000 10010000                            4 bits/genotype
          10000000 00000010
          00000000 00000000
```

Run-Length Encoding Complicates* Logical Operations

```
      00000000 00000111 11111111 11110000
  OR  11111111 10000000 00000000 00000000
  -----------------------------------------
      11111111 10000111 11111111 11111111
```



13 bits

```
      00001101 10001111 00000100
  OR  10001001 10010111
  -----------------------------
```

9 bits

Succinct Data Structure

- Near-optimal compression
- Allow operations without inflation

*likely, but not proven to be impractical

Word Aligned Hybrid [Wu, TODS 2006]

Word Aligned Hybrid [Wu, TODS 2006]

Run Length Encoding

value : length (bits)

```
10010000  →  1:1       0:2       1:1       0:4
             10000001  00000010  10000001  00000100
```

Word Aligned Hybrid

Literal

type : uncompressed value

```
1001000  →  0101000
```

Fill

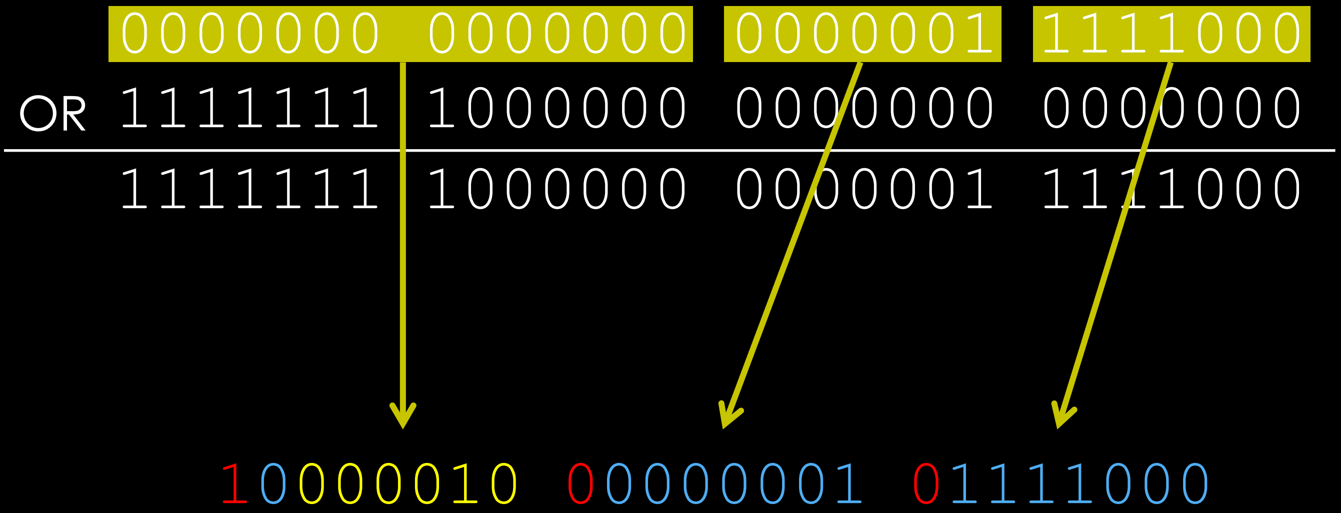
type : value : length (words)

```
0000000 0000000          →  1000010
1111111 1111111 1111111  →  1100011
```

Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
   -------------------------------
   1111111 1000000 0000001 1111000
```

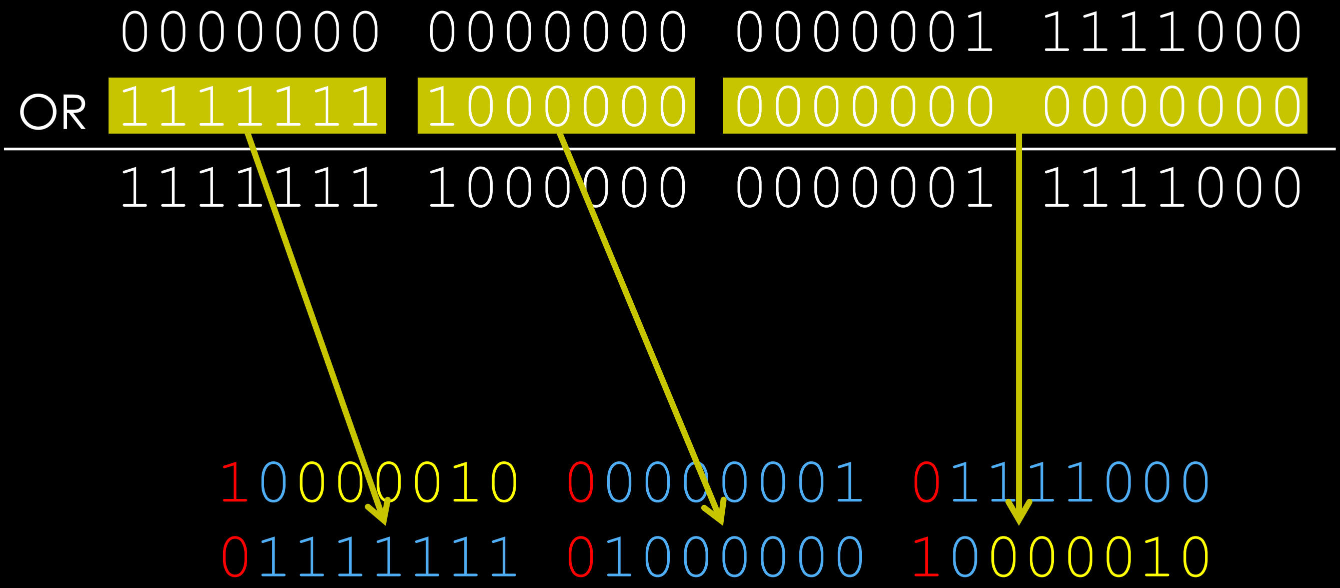
Word Aligned Hybrid [Wu, TODS 2006]



```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
   -------------------------------
   1111111 1000000 0000001 1111000

     10000010 00000001 01111000
```

Word Aligned Hybrid [Wu, TODS 2006]



```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
   1111111 1000000 0000001 1111000

   10000010 00000001 01111000
   01111111 01000000 10000010
```

Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
-----------------------------------
   1111111 1000000 0000001 1111000
```

```
10000010 00000001 01111000
01111111 01000000 10000010
```

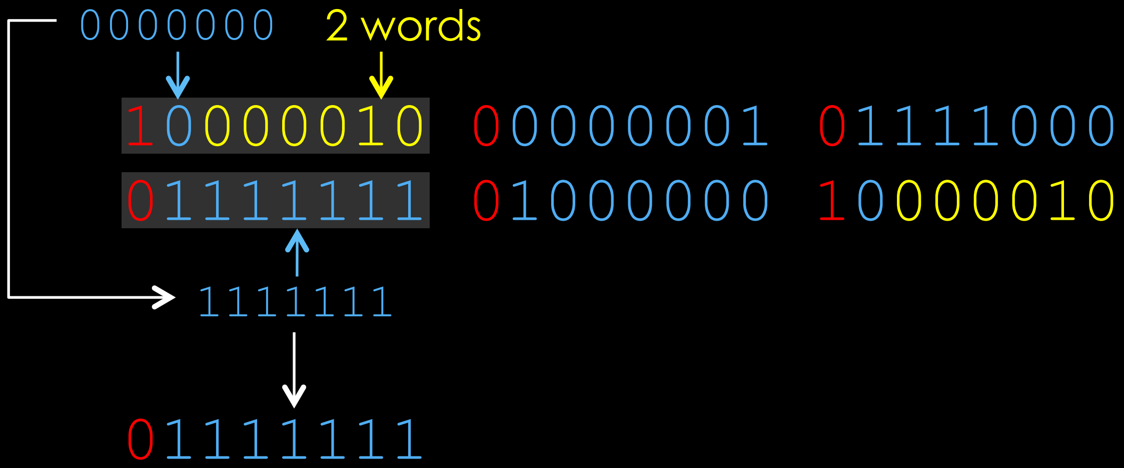
Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
   1111111 1000000 0000001 1111000
```

```
10000010 00000001 01111000
01111111 01000000 10000010
```

Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
----------------------------------
   1111111 1000000 0000001 1111000
```



0000000 2 words

10000010 00000001 01111000

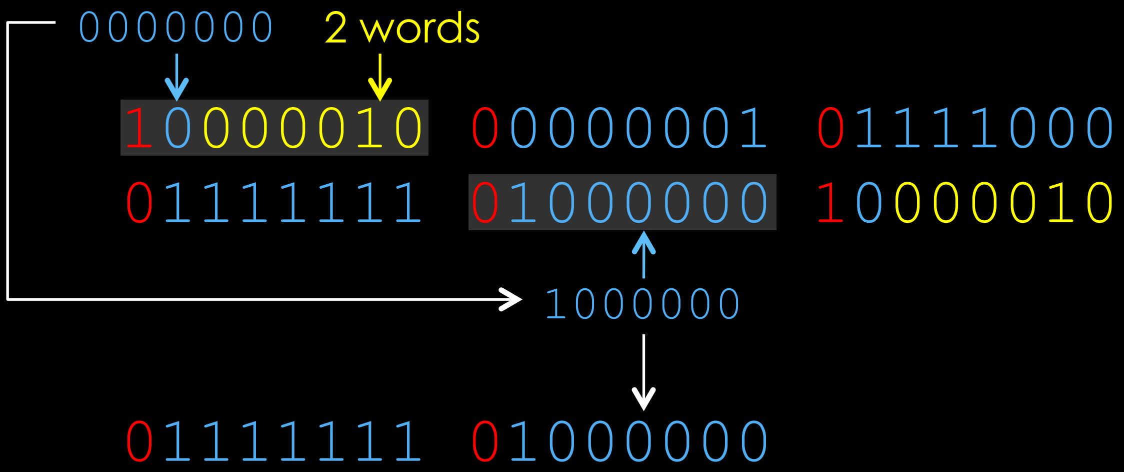
01111111 01000000 10000010

1111111

01111111

Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
   -------------------------------
   1111111 1000000 0000001 1111000
```



0000000

2 words

10000010 00000001 01111000

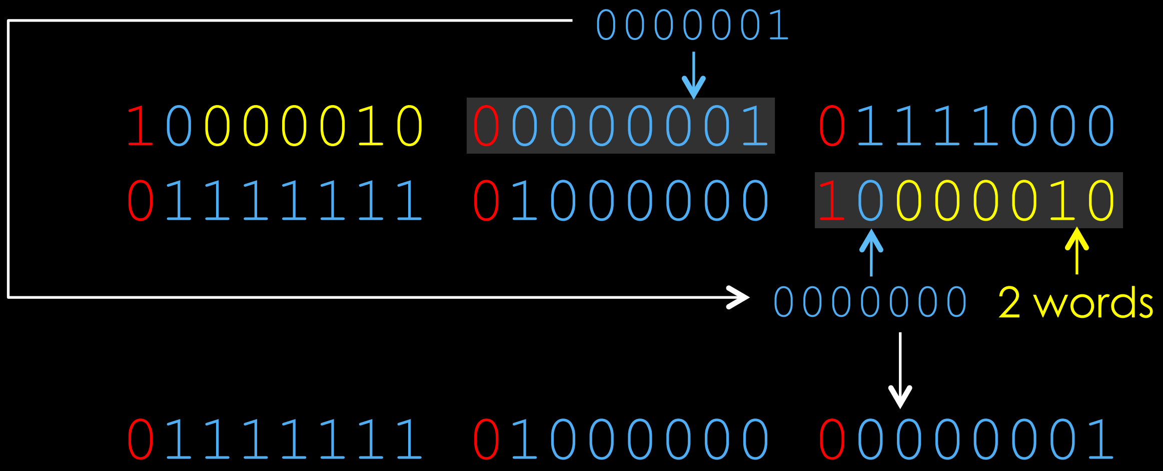
01111111 01000000 10000010

1000000

01111111 01000000

Word Aligned Hybrid [Wu, TODS 2006]

```
     0000000 0000000 0000001 1111000
OR   1111111 1000000 0000000 0000000
-------------------------------------
     1111111 1000000 0000001 1111000
```



0000001

10000010 00000001 01111000

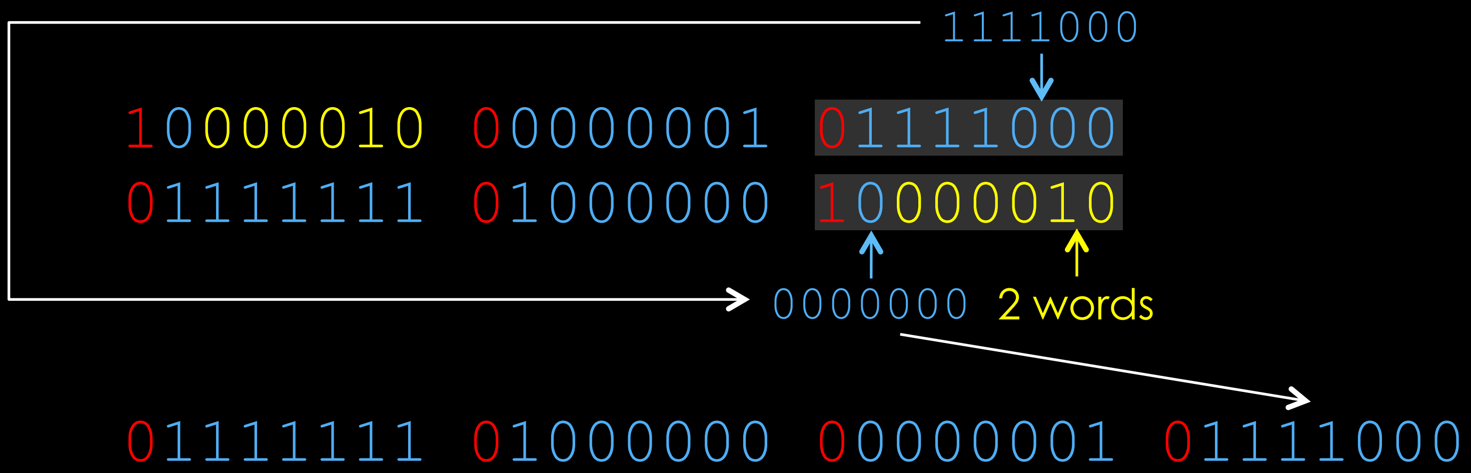
01111111 01000000 10000010

0000000 2 words

01111111 01000000 00000001

Word Aligned Hybrid [Wu, TODS 2006]

```
   0000000 0000000 0000001 1111000
OR 1111111 1000000 0000000 0000000
----------------------------------
   1111111 1000000 0000001 1111000
```



1111000

10000010 00000001 01111000

01111111 01000000 10000010

0000000 2 words

01111111 01000000 00000001 01111000

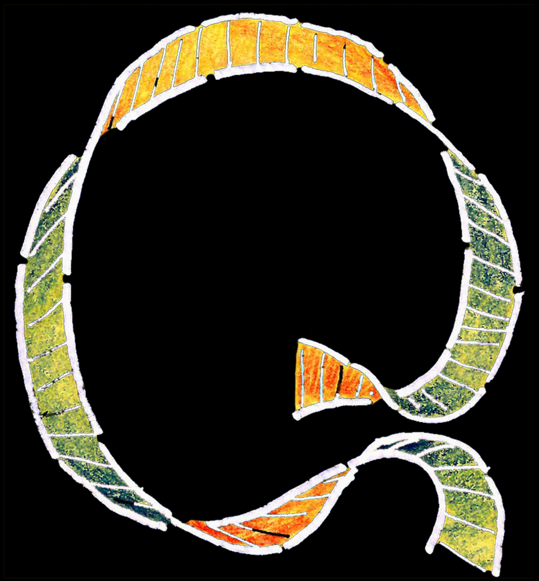
Word Aligned Hybrid [Wu, TODS 2006]

Smaller Files:

Compresses with run-length and literal values

Faster Queries:

Bitwise logical operations without inflation



GENOTYPE QUERY TOOLS

Tool and C API for large-scale genotype queries

Allele frequency sorting

WAH encoding

`https://github.com/ryanlayer/gqt`

MACS simulation

1e8 Genome

100 ... 100000 individuals

588830 ... 2061134 variants

1000 Genomes chr15 validation

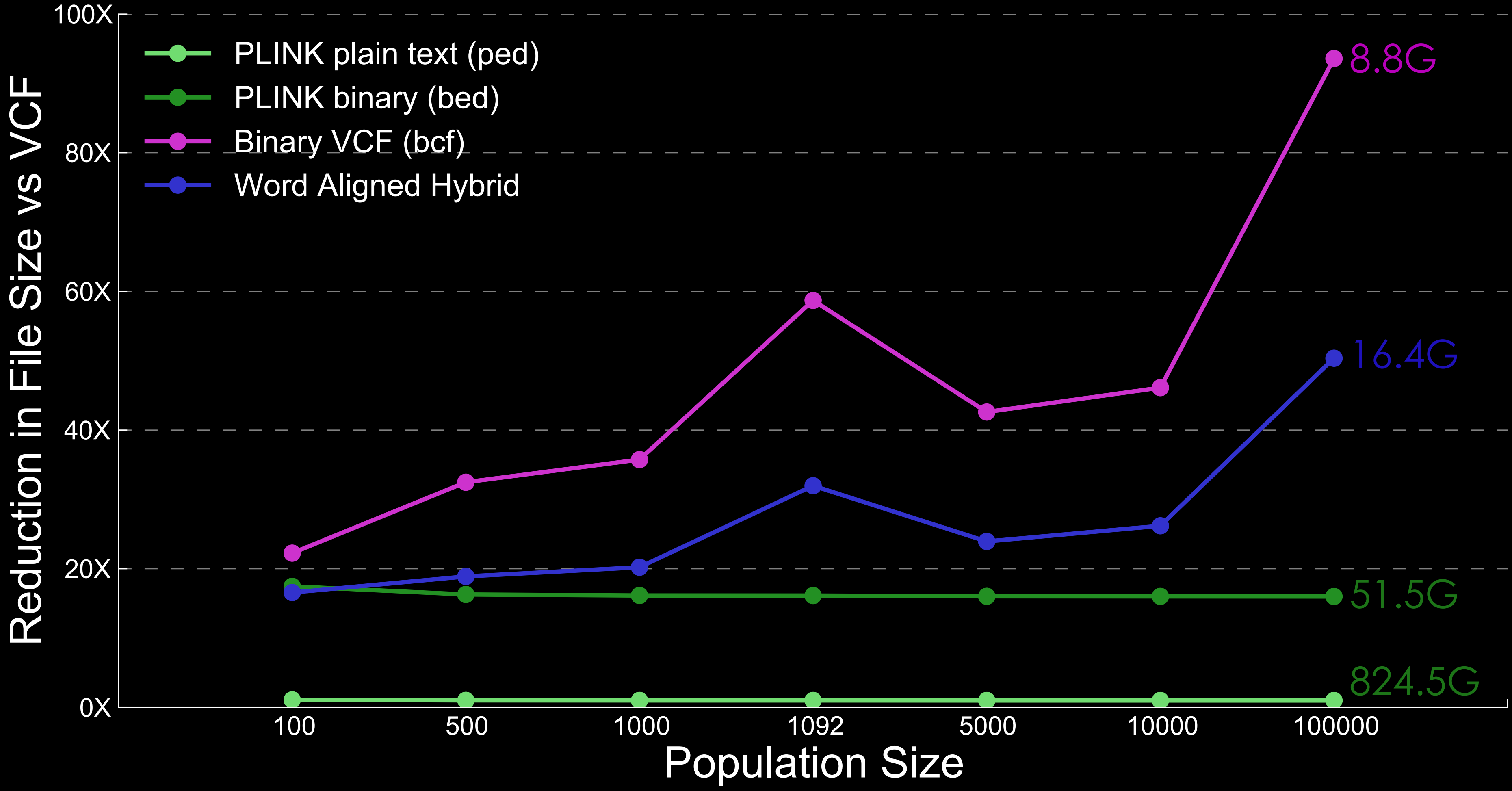
	1KG	Simulation	Difference
genome size	102531392	100000000	0.98
individuals	1092	1000	0.92
variants	1130554	816284	0.72

File size and query time wrt bcftools, plink, plink2

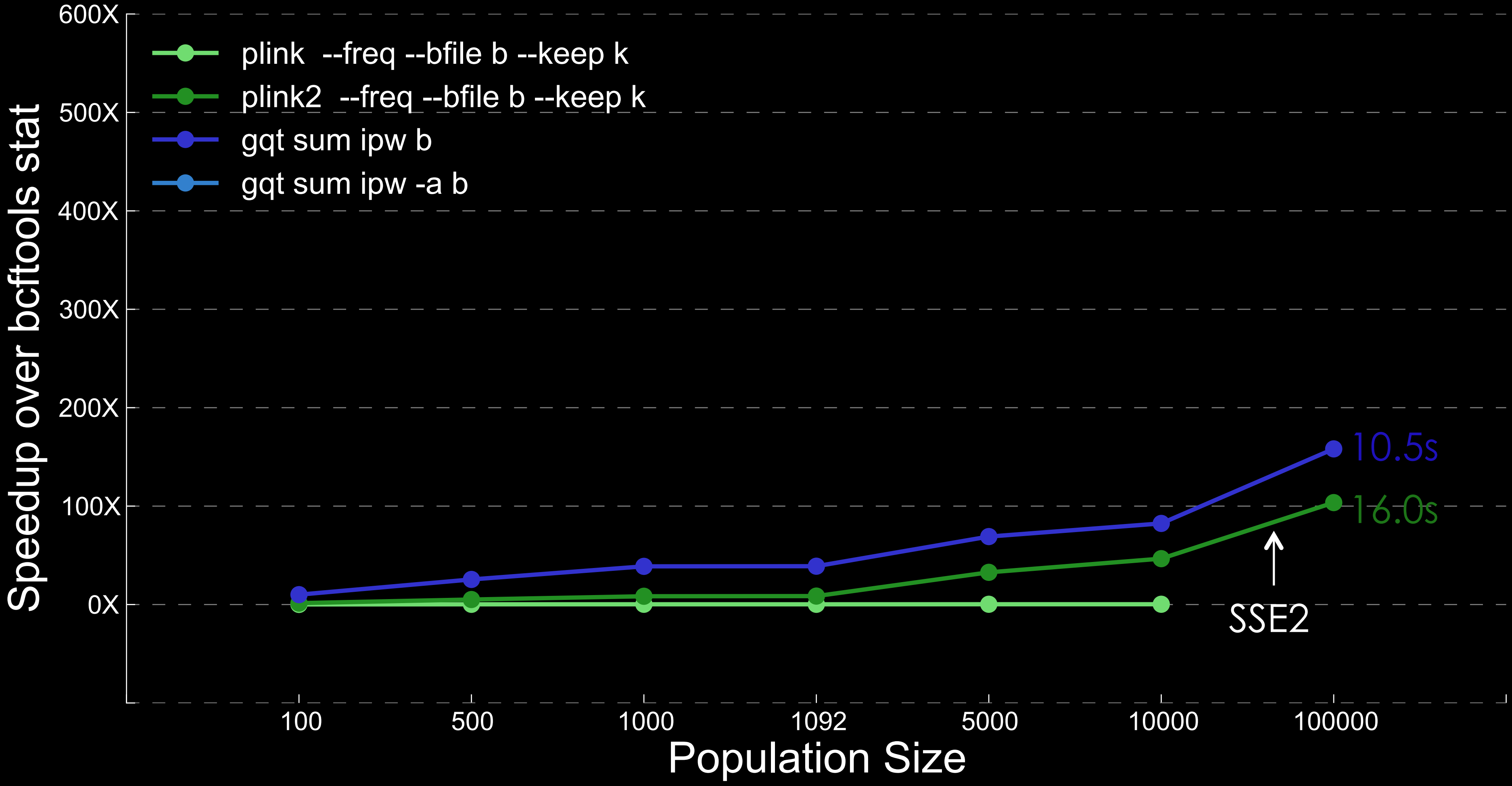
```
https://github.com/samtools/bcftools
http://pngu.mgh.harvard.edu/~purcell/plink/
https://github.com/chrchang/plink-ng
```

MacBook Pro, 2.8GHz Intel Core i7 (Haswell)

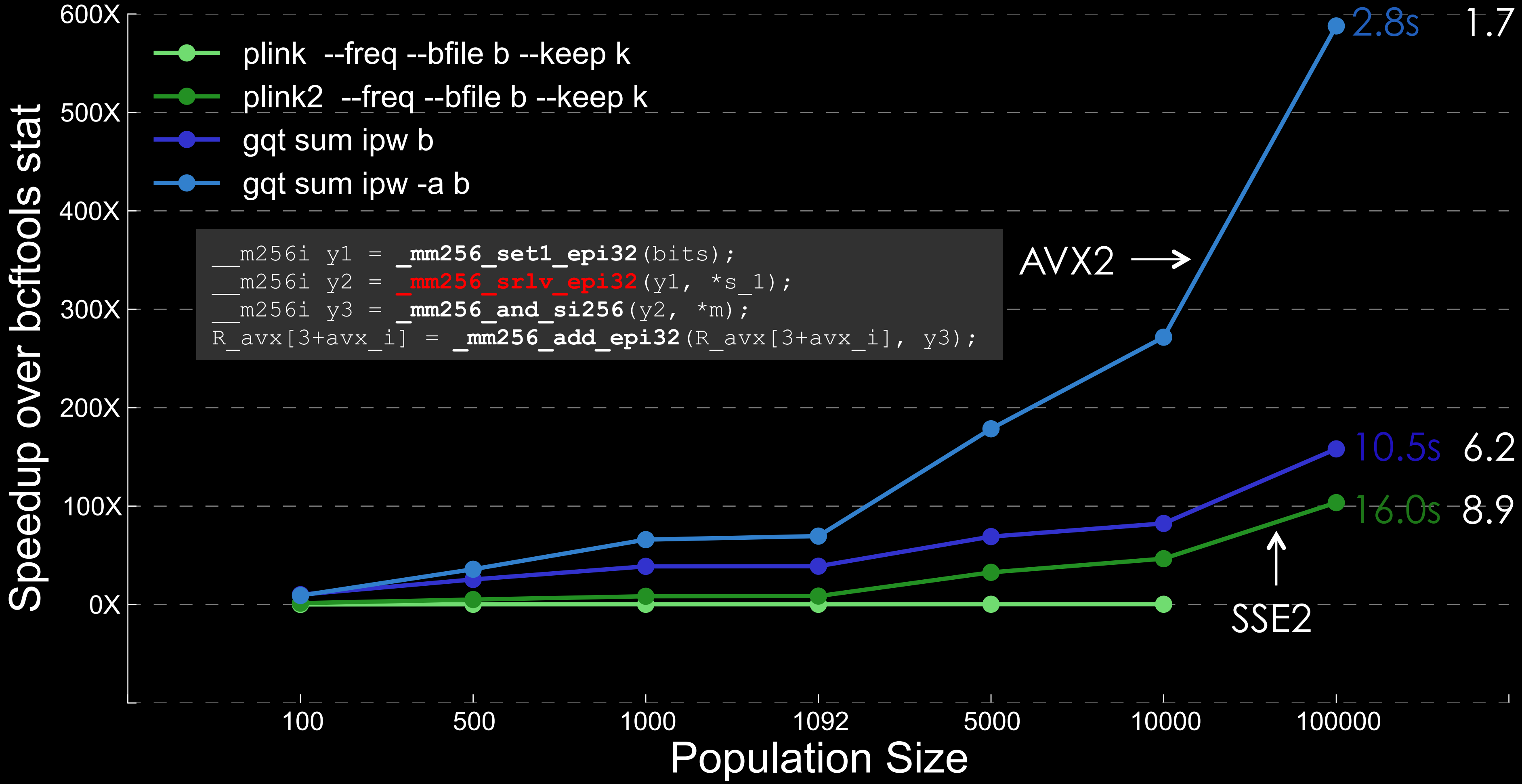
Encoded File Size vs. VCF



Speedup for Alt. Allele Count in 10% of Population vs. bcftools stat -s

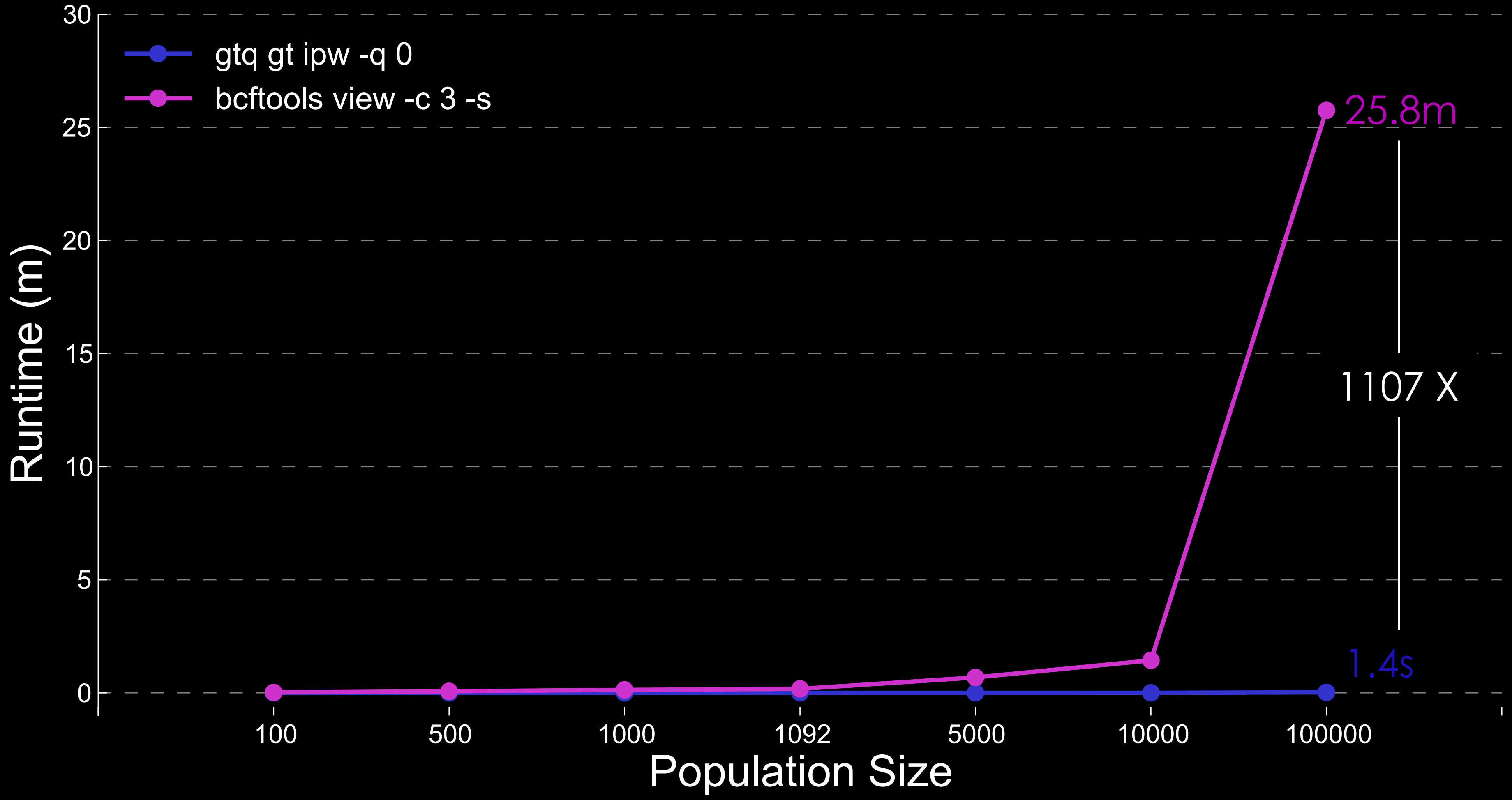


Speedup for Alt. Allele Count in 10% of Population vs. bcftools stat -s

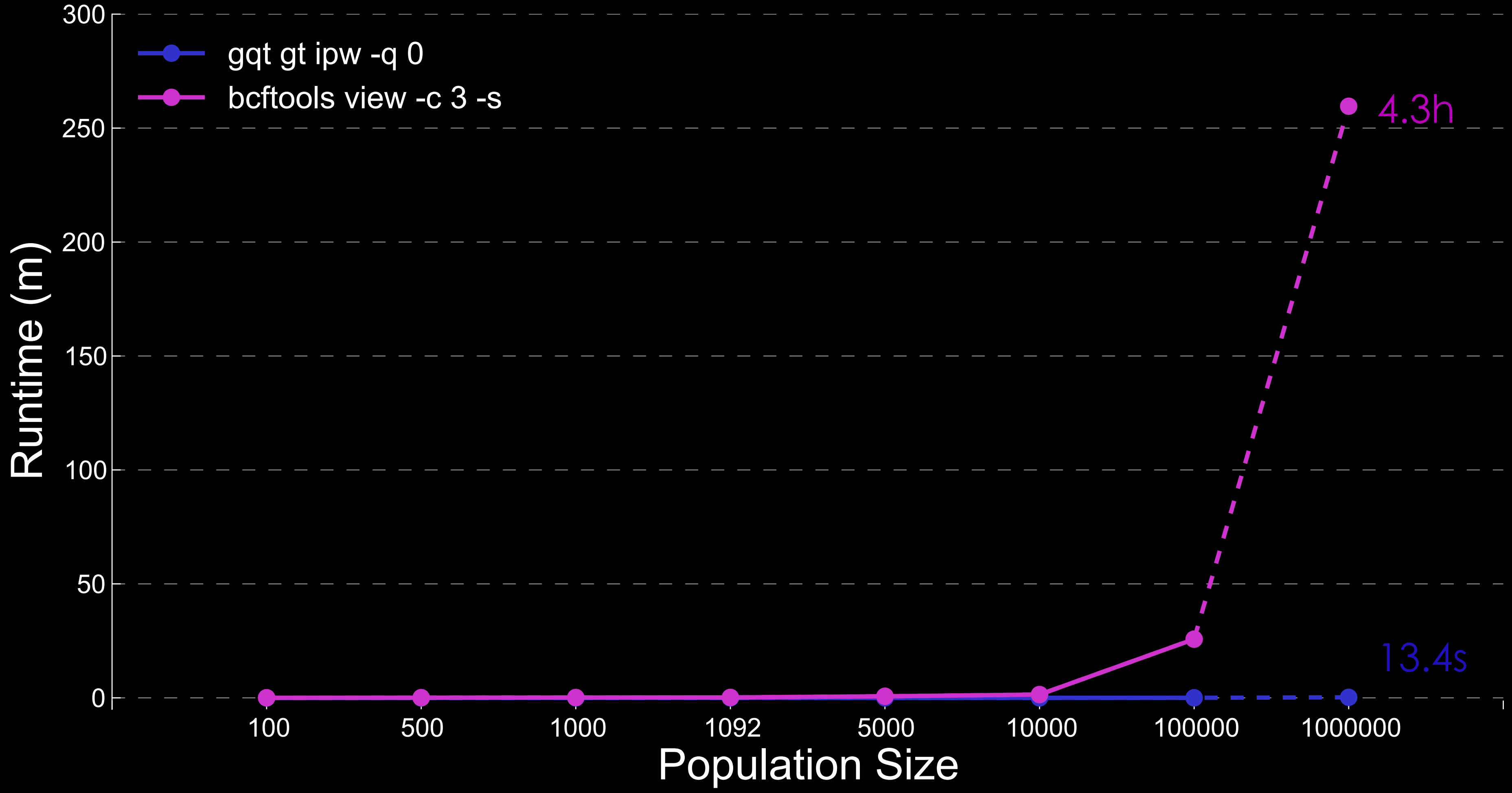


```
__m256i y1 = _mm256_set1_epi32(bits);
__m256i y2 = _mm256_srlv_epi32(y1, *s_1);
__m256i y3 = _mm256_and_si256(y2, *m);
R_avx[3+avx_i] = _mm256_add_epi32(R_avx[3+avx_i], y3);
```

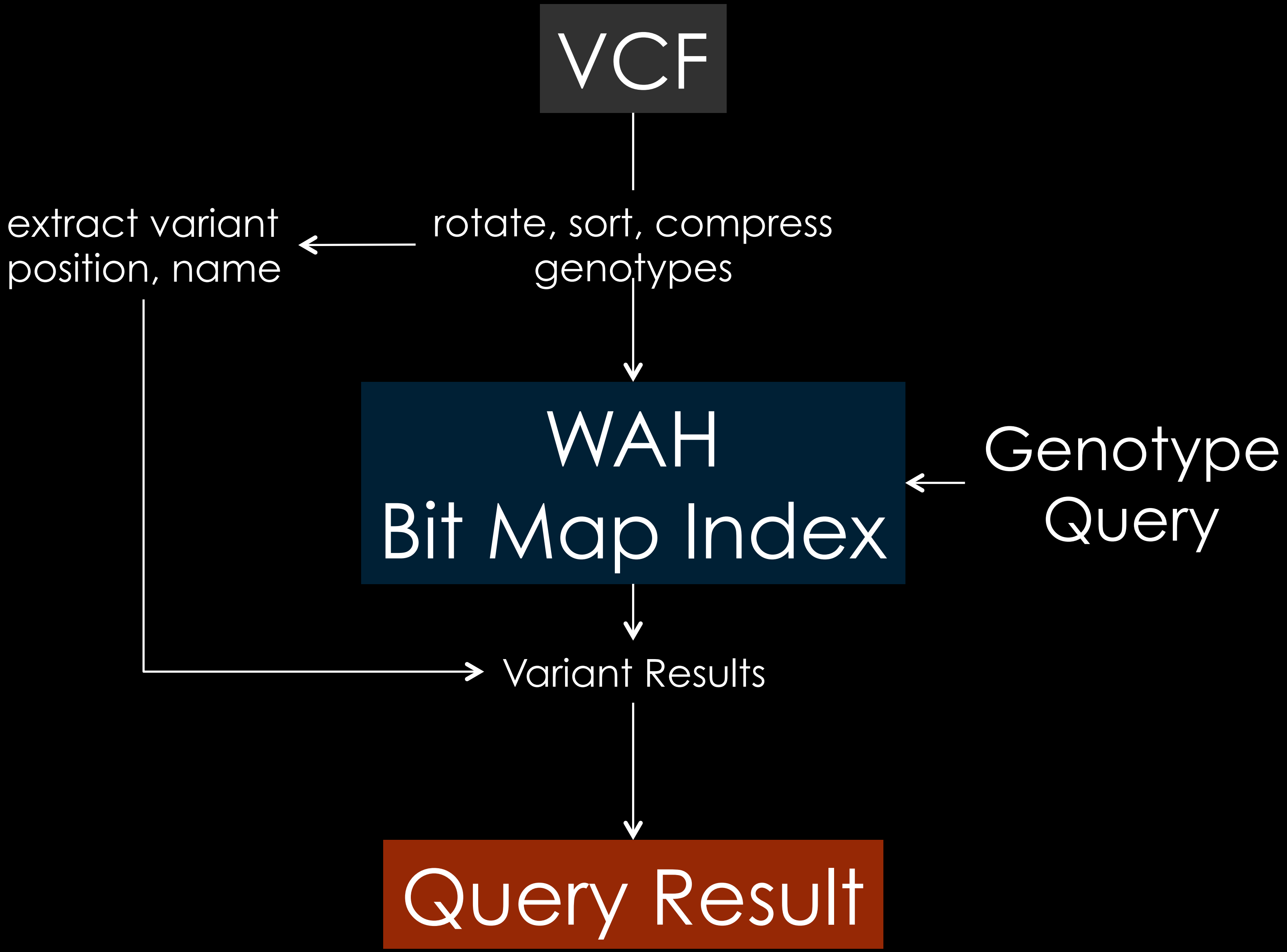
Runtime for Finding Non-Reference Variants in 10% of the Population



1 Million Individuals (Linear Fit Estimate)



Current System



VCF

rotate, sort, compress genotypes

extract variant position, name

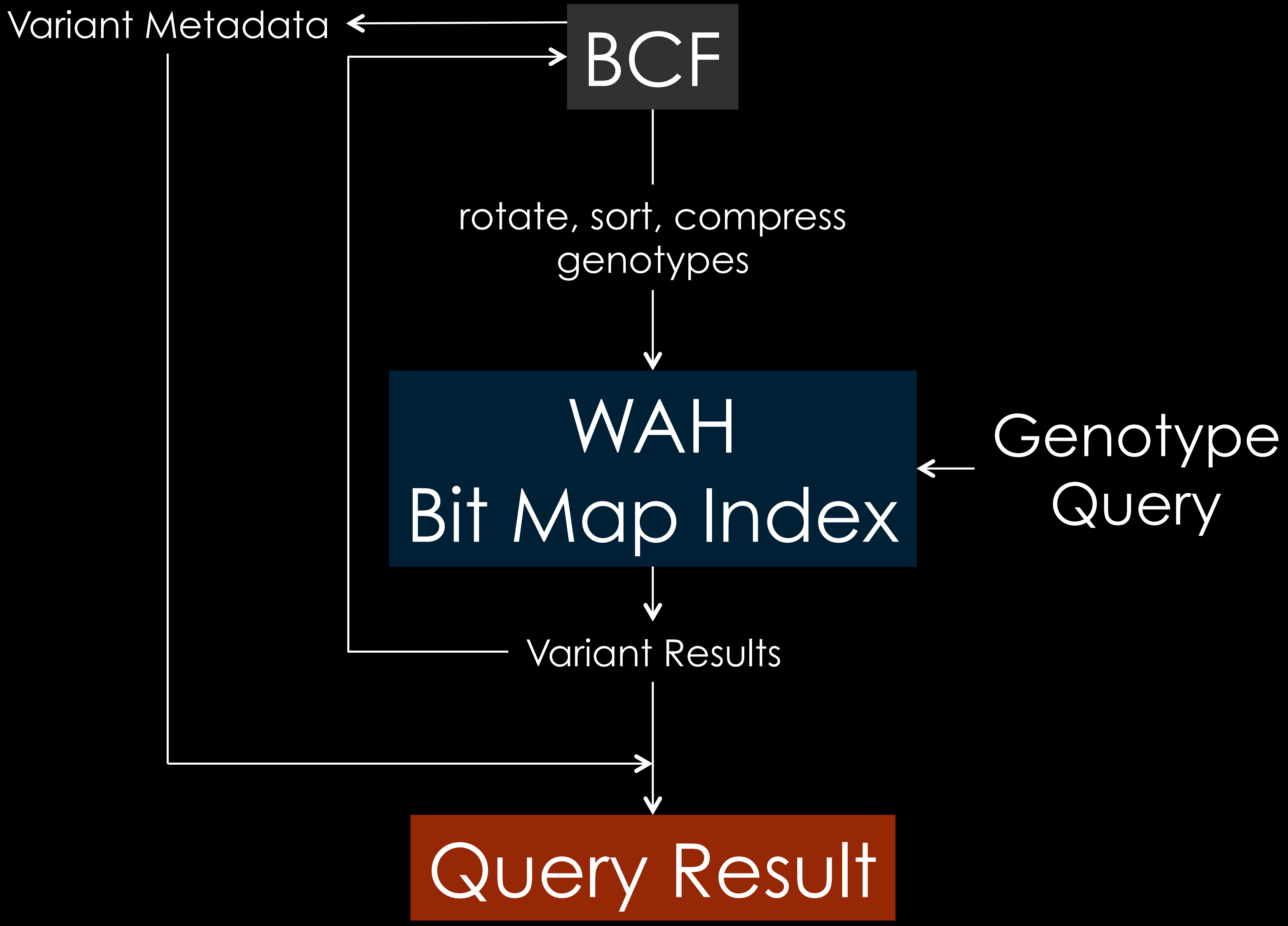
WAH Bit Map Index

Genotype Query

Variant Results

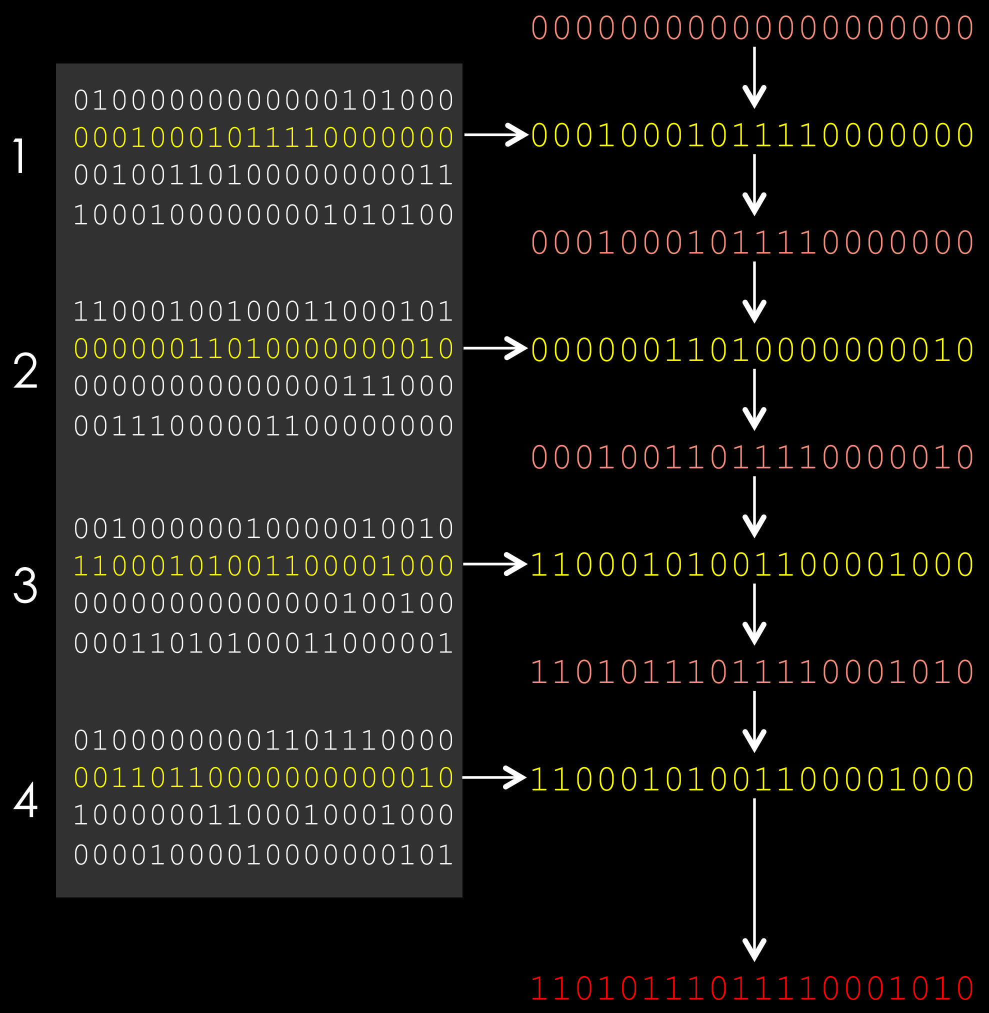
Query Result

Envisioned System



Gemini Query Engine

Single Core Sequential Processing



1

01000000000000101000
00010001011110000000
00100110100000000011
10001000000001010100

2

11000100100011000101
00000011010000000010
00000000000000111000
00111000001100000000

3

00100000010000010010
11000101001100001000
00000000000000100100
00011010100011000001

4

01000000001101110000
00110110000000000010
10000001100010001000
00001000010000000101

00000000000000000000
00010001011110000000
00010001011110000000
00000011010000000010
00010011011110000010
11000101001100001000
11010111011110001010
11000101001100001000
11010111011110001010

Distributed Parallel Processing

1

```
01000000000000101000
00010001011110000000
00100110100000000011
10001000000001010100
```

2

```
11000100100011000101
00000011010000000010
00000000000000111000
00111000001100000000
```

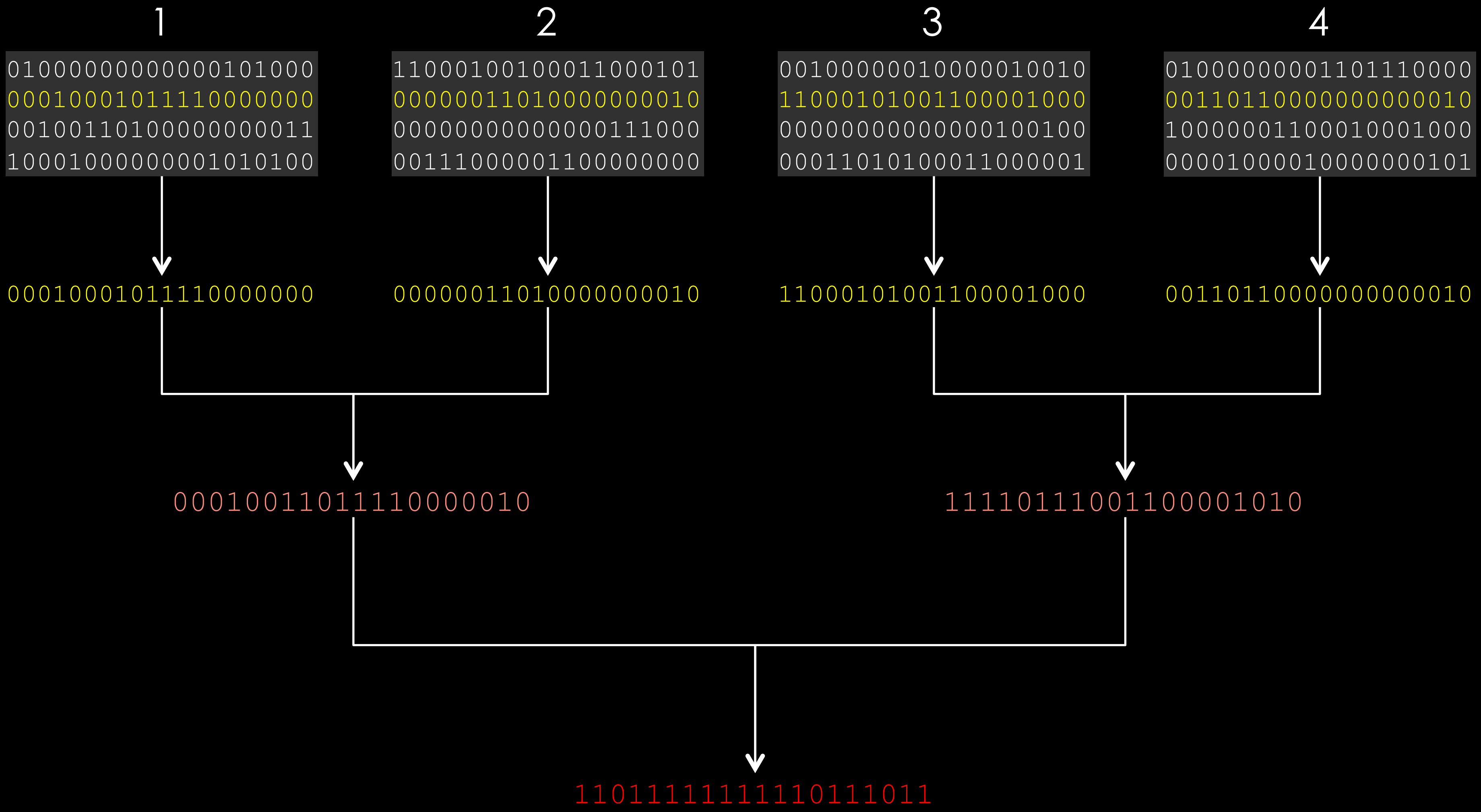
3

```
00100000010000010010
11000101001100001000
00000000000000100100
00011010100011000001
```

4

```
01000000001101110000
00110110000000000010
10000001100010001000
00001000010000000101
```

Distributed Parallel Processing



1

01000000000000101000
00010001011110000000
00100110100000000011
10001000000001010100

2

11000100100011000101
00000011010000000010
00000000000000111000
00111000001100000000

3

00100000010000010010
11000101001100001000
00000000000000100100
00011010100011000001

4

01000000001101110000
00110110000000000010
10000001100010001000
00001000010000000101

00010001011110000000

00000011010000000010

11000101001100001000

00110110000000000010

00010011011110000010

11110111001100001010

11011111111110111011

Genotype Query tools

https://github.com/ryanlayer/gqt

rl6sf@virginia.edu

@ryanlayer

Millions of Individuals

100 Millions of genotypes

Rotate, Sort, WAH Encode

Small Files

Fast Queries

Aaron Quinlan

Neil Kindlon

NIH Cancer Training Grant postdoc fellowship