Lecture 4: Sequencing Coverage Analysis + Introduction to Genome-Wide Association Studies (GWAS)

ECE 365 - Data Science and Genomics

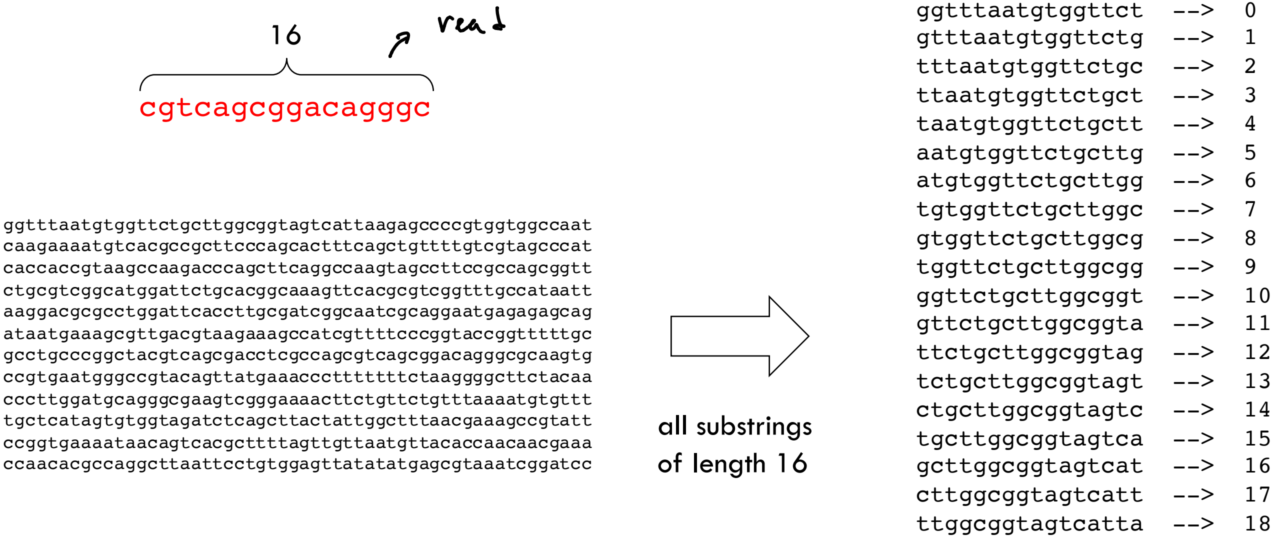
Announcements:

- Lab 1 due today (11:59pm)
- Lab 2 released (due March 18)

Today:

- Finish discussion on sequencing coverage
- Introduction to Genome-Wide Association Studies (GWAS)

Indexing



```
cgtcagcggacagggc
```

```
ggtttaatgtggttctgcttggcggtagtcattaagagccccgtggtggccaat
caagaaaatgtcacgccgcttcccagcactttcagctgttttgtcgtagcccat
caccaccgtaagccaagacccagcttcaggccaagtagccttccgccagcggtt
ctgcgtcggcatggattctgcacggcaaagttcacgcgtcggtttgccataatt
aaggacgcgcctggattcaccttgcgatcggcaatcgcaggaatgagagagcag
ataatgaaagcgttgacgtaagaaagccatcgttttcccggtaccggtttttgc
gcctgcccggctacgtcagcgacctcgccagcgtcagcggacagggcgcaagtg
ccgtgaatgggccgtacagttatgaaaccctttttttctaagggcttctacaa
cccttggatgcagggcgaagtcgggaaaacttctgttctgtttaaaatgtgttt
tgctcatagtgtggtagatctcagcttactattggctttaacgaaagccgtatt
ccggtgaaaataacagtcacgcttttagttgttaatgttacaccaacaacgaaa
ccaacacgccaggcttaattcctgtggagttatatatgagcgtaaatcggatcc
```

```
ggtttaatgtggttct  -->  0
gtttaatgtggttctg  -->  1
tttaatgtggttctgc  -->  2
ttaatgtggttctgct  -->  3
taatgtggttctgctt  -->  4
aatgtggttctgcttg  -->  5
atgtggttctgcttgg  -->  6
tgtggttctgcttggc  -->  7
gtggttctgcttggcg  -->  8
tggttctgcttggcgg  -->  9
ggttctgcttggcggt  -->  10
gttctgcttggcggta  -->  11
ttctgcttggcggtag  -->  12
tctgcttggcggtagt  -->  13
ctgcttggcggtagtc  -->  14
tgcttggcggtagtca  -->  15
gcttggcggtagtcat  -->  16
cttggcggtagtcatt  -->  17
ttggcggtagtcatta  -->  18
```

- □ Another way to do indexing: Python dictionary (hash function)
- □ Let’s look at this in a notebook

What if there are errors/mutations on read?

$X =$ cgt$\overset{\downarrow}{\text{a}}$agcggaca$\overset{\downarrow}{\text{t}}$ggc

$Y =$

ggtttaatgtggttctgcttggcggtagtcattaagagccccgtggtggccaat
caagaaaatgtcacgccgcttcccagcactttcagctgttttgtcgtagcccat
caccaccgtaagccaagacccagcttcaggccaagtagccttccgccagcggtt
ctgcgtcggcatggattctgcacggcaaagttcacgcgtcggtttgccataatt
aaggacgcgcctggattcaccttgcgatcggcaatcgcaggaatgagagagcag
ataatgaaagcgttgacgtaagaaagccatcgttttcccggtaccggtttttgc
gcctgcccggctacgtcagcgacctcgccag**cgtcagcggacagggc**gcaagtg
ccgtgaatgggccgtacagttatgaaacccttttttttctaaggggcttctacaa
cccttggatgcagggcgaagtcgggaaaacttctgttctgtttaaaatgtgttt
tgctcatagtgtggtagatctcagcttactattggctttaacgaaagccgtatt
ccggtgaaaataacagtcacgcttttagttgttaatgttacaccaacaacgaaa
ccaacacgccaggcttaattcctgtggagttatatatgagcgtaaatcggatcc

What if there are errors/mutations on read?

$X =$ cgt**a**agcggaca**t**ggc

$Y =$

```
ggtttaatgtggttctgcttggcggtagtcattaagagccccgtggtggccaat
caagaaaatgtcacgccgcttcccagcactttcagctgttttgtcgtagcccat
caccaccgtaagccaagacccagcttcaggccaagtagccttccgccagcggtt
ctgcgtcggcatggattctgcacggcaaagttcacgcgtcggtttgccataatt
aaggacgcgcctggattcaccttgcgatcggcaatcgcaggaatgagagagcag
ataatgaaagcgttgacgtaagaaagccatcgttttcccggtaccggtttttgc
gcctgcccggctacgtcagcgacctcgccagcgtcagcggacagggcgcaagtg
ccgtgaatgggccgtacagttatgaaaccctttttttctaaggggcttctacaa
cccttggatgcagggcgaagtcgggaaaacttctgttctgtttaaaatgtgttt
tgctcatagtgtggtagatctcagcttactattggctttaacgaaagccgtatt
ccggtgaaaataacagtcacgcttttagttgttaatgttacaccaacaacgaaa
ccaacacgccaggcttaattcctgtggagttatatatgagcgtaaatcggatcc
```

- □ One idea: Consider indexing substrings of length k ≈ 6

What if there are errors/mutations on read?

$X =$ cgtaagcggacatggc

$Y =$

```
ggtttaatgtggttctgcttggcggtagtcattaagagccccgtggtggccaat
caagaaaatgtcacgccgcttcccagcactttcagctgttttgtcgtagcccat
caccaccgtaagccaagacccagcttcaggccaagtagccttccgccagcggtt
ctgcgtcggcatggattctgcacggcaaagttcacgcgtcggtttgccataatt
aaggacgcgcctggattcaccttgcgatcggcaatcgcaggaatgagagagcag
ataatgaaagcgttgacgtaagaaagccatcgttttcccggtaccggtttttgc
gcctgcccggctacgtcagcgacctcgccagcgtcagcggacagggcgcaagtg
ccgtgaatgggccgtacagttatgaaacccttttttttctaagggcttctacaa
cccttggatgcagggcgaagtcgggaaaacttctgttctgtttaaaatgtgttt
tgctcatagtgtggtagatctcagcttactattggctttaacgaaagccgtatt
ccggtgaaaataacagtcacgcttttagttgttaatgttacaccaacaacgaaa
ccaacacgccaggcttaattcctgtggagttatatatgagcgtaaatcggatcc
```

all substrings of length $k = 6$

Python Dict ()

```
ggttta --> [0]
gtttaa --> [1, 471]
tttaat --> [2]
ttaatg --> [3, 571]
taatgt --> [4, 572]
aatgtg --> [5, 477]
atgtgg --> [6]
tgtggt --> [7, 495]
gtggtt --> [8]
tggttc --> [9]
ggttct --> [10, 158]
gttctg --> [11, 159, 466]
ttctgc --> [12, 160, 177]
tctgct --> [13]
ctgctt --> [14]
tgcttg --> [15]
gcttgg --> [16]
cttggc --> [17]
ttggcg --> [18]
tggcgg --> [19]
ggcggt --> [20]
```

- □ One idea: Consider indexing substrings of length k

What if there are errors/mutations on read?

$X =$ cgtaagcggacatggc

take substrings
of length $k = 6$

```
cgtaag --> [114, 286]
gtaagc --> [115]
taagcg -->
aagcgg -->
agcgga --> [359]
gcggac --> [360]
cggaca --> [361]
ggacat -->
gacatg -->
acatgg -->
catggc -->
```

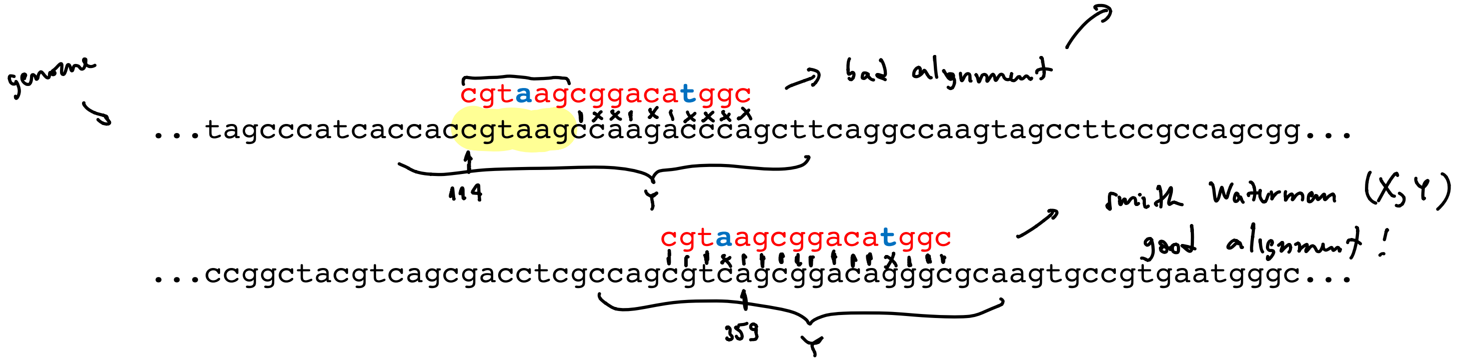
What if there are errors/mutations on read?

$X =$ cgtaagcggacatggc

take substrings
of length $k = 6$

```
cgtaag --> [114, 286]
gtaagc --> [115]
taagcg -->
aagcgg -->
agcgga --> [359]
gcggac --> [360]
cggaca --> [361]
ggacat -->
gacatg -->
acatgg -->
catggc -->
```

Smith Waterman (X, Y)



```
            cgtaagcggacatggc
...tagcccatcaccaccgtaagccaagacccagcttcaggccaagtagccttccgccagcgg...

                              cgtaagcggacatggc
...ccggctacgtcagcgacctcgccagcgtcagcggacagggcgcaagtgccgtgaatgggc...
```

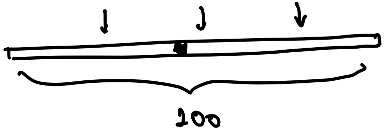
How do we choose k?

$\lceil 3.2 \rceil = 4$

$\lceil 5 \rceil = 5$

□ Based on the sequencing technology (e.g., Illumina: $L = 100$, error rate $= 0.1\%$)

- Suppose at most 1 error per read



Claim: there is a segment of length $\frac{100 - 1}{2} = 49.5$ with no errors

- Suppose $\leq t$ errors per read

there is a segment of length $\underbrace{\left\lceil \frac{L - t}{t+1} \right\rceil}$ with no errors

↪ set $= k$

How many reads have $> t$ errors?

# of errors in a read : $X \sim \text{Binomial}(100, 0.001)$

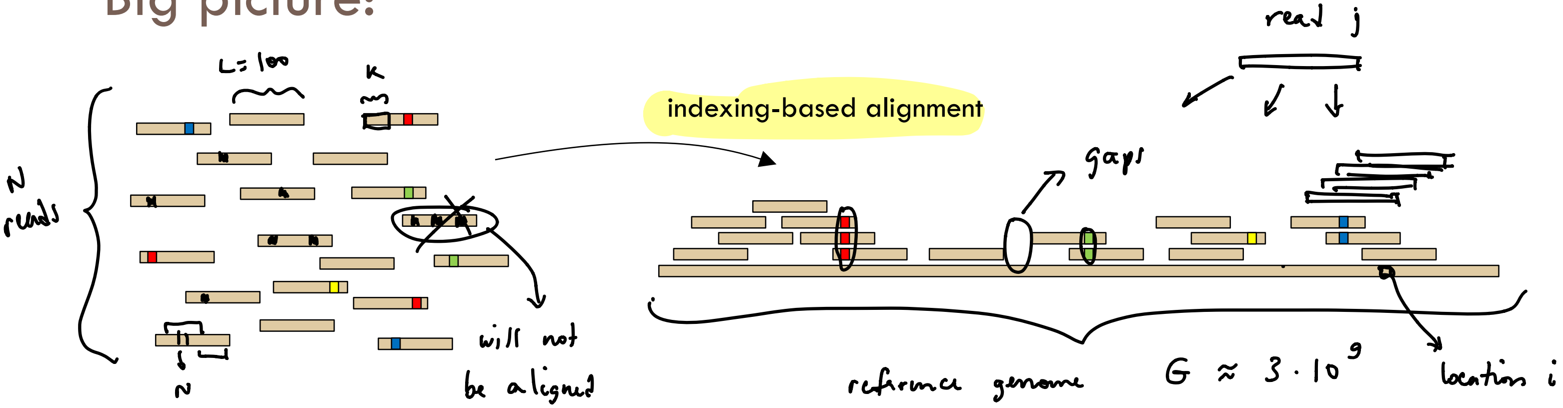
fraction of reads with $> t$ errors: $P(X > t)$

e.g. for $t = 2$.

$P(X > 2) = 0.00015$

$k = \left\lceil \frac{100 - 2}{3} \right\rceil = 33$

Big picture:


indexing-based alignment

How to pick N ?

Assume that start location of each read is uniformly distributed on genome

$P(\text{read } j \text{ covers position } i) = \frac{L}{G}$

$\Rightarrow$ Expected # of reads covering position i $= \frac{NL}{G} \triangleq c$ (coverage depth)

Prob. that there are gaps? Fix i.

$$P(i \text{ is uncovered}) = \prod_{j=1}^{N}\left(1-\frac{L}{G}\right) = \left(1-\frac{L}{G}\right)^{N} = \underbrace{\left(1-\frac{1}{\frac{G}{L}}\right)^{\frac{G}{L}}}_{\approx e^{-1}}{}^{\cdot\underbrace{\frac{NL}{G}}_{c}} \approx e^{-c}$$

Recall: $\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n} = e^{-1}$

union bound (good approximation)

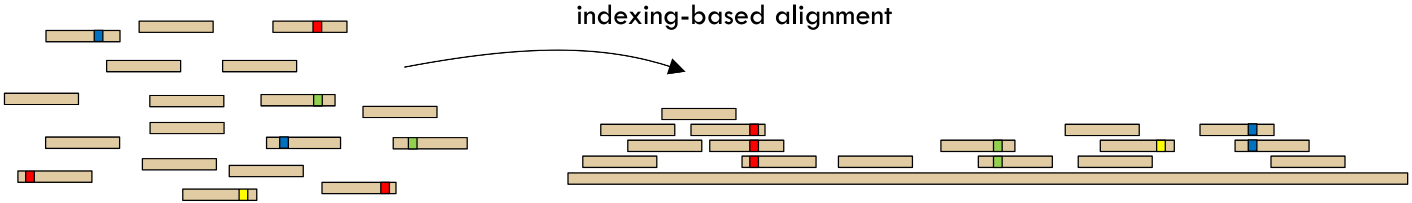
$$P(\text{some position is uncovered}) \lesssim G e^{-c} = \varepsilon \quad (\text{desired failure prob.})$$

$$\Rightarrow \quad e^{c} = \frac{G}{\varepsilon} \quad \Rightarrow \quad c = \ln\left(\frac{G}{\varepsilon}\right)$$

Since $c = \frac{NL}{G}$, $N = \frac{G}{L}\ln\left(\frac{G}{\varepsilon}\right)$. (Lander - Waterman)

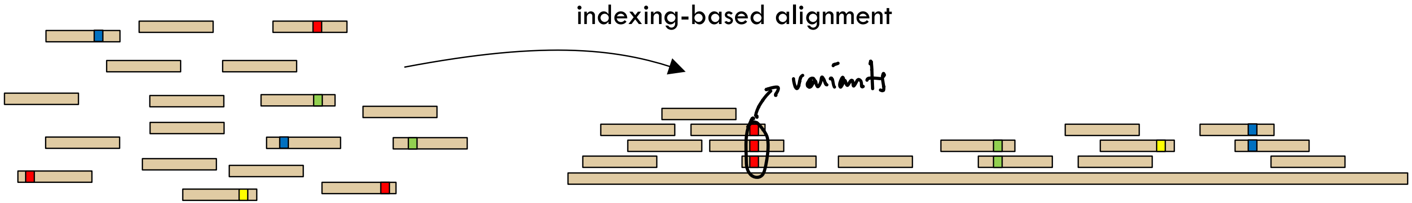
Big picture:

- Map sequencing data to a reference genome



Big picture:

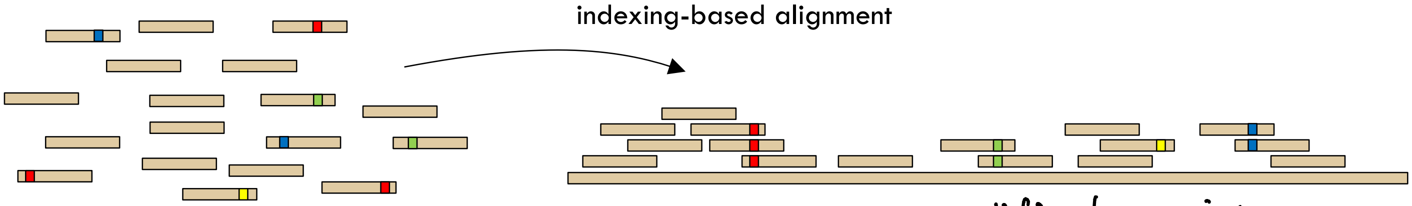
- □ Map sequencing data to a reference genome



- □ Using aligned data, we can identify *variants*

Big picture:

- □ Map sequencing data to a reference genome

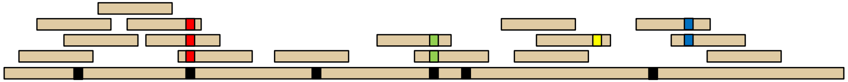


- □ Using aligned data, we can identify *variants*
- □ Variants allow us to create genotype matrices

$$\begin{array}{rc} & \overbrace{\hphantom{0\quad 1\quad 1\quad 0\quad 0\quad 1}}^{\text{different variants}} \\ \begin{array}{r} \text{person 1} \\ \\ \\ \vdots \\ \\ \\ \text{person } n \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ & \text{genotype} \end{array}$$

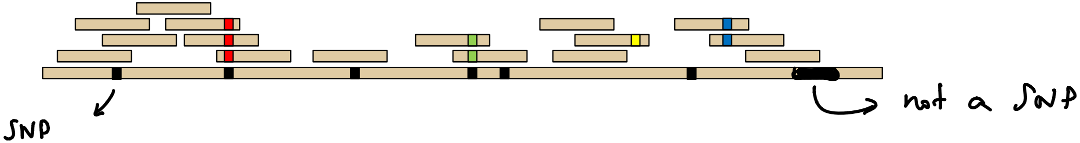
Genotype data

- Focus on a set of common variants in the genome



Genotype data

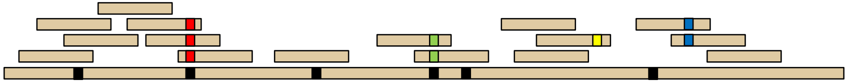
- □ Focus on a set of common variants in the genome



- □ Typically, we focus on Single-Nucleotide Polymorphisms (SNPs, read snips)

Genotype data

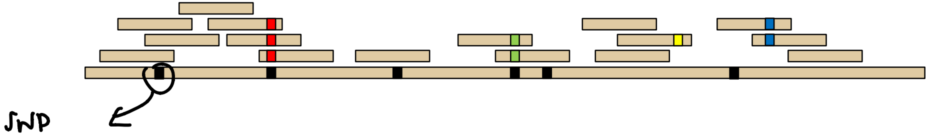
- □ Focus on a set of common variants in the genome



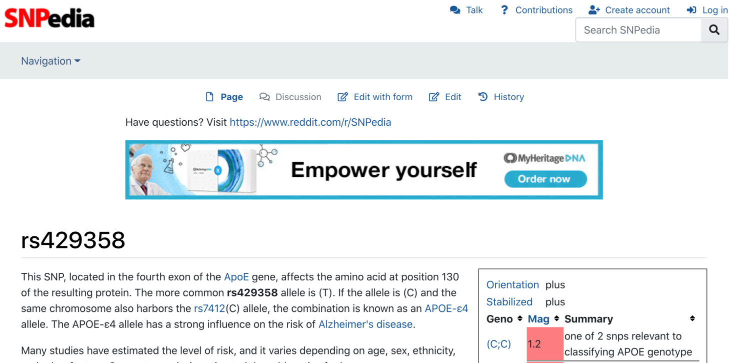
- □ Typically, we focus on Single-Nucleotide Polymorphisms (SNPs, read snips)
- □ Most SNPs only allow two possible values: REF and ALT (e.g. REF = A, ALT = C)

Genotype data

- □ Focus on a set of common variants in the genome



- □ Typically, we focus on Single-Nucleotide Polymorphisms (SNPs, read snips)
- □ Most SNPs only allow two possible values: REF and ALT
- □ More info on SNPs: SNPedia!
  - Search for rs429358



Have questions? Visit https://www.reddit.com/r/SNPedia

rs429358

This SNP, located in the fourth exon of the ApoE gene, affects the amino acid at position 130 of the resulting protein. The more common **rs429358** allele is (T). If the allele is (C) and the same chromosome also harbors the rs7412(C) allele, the combination is known as an APOE-ε4 allele. The APOE-ε4 allele has a strong influence on the risk of Alzheimer's disease.

Many studies have estimated the level of risk, and it varies depending on age, sex, ethnicity,

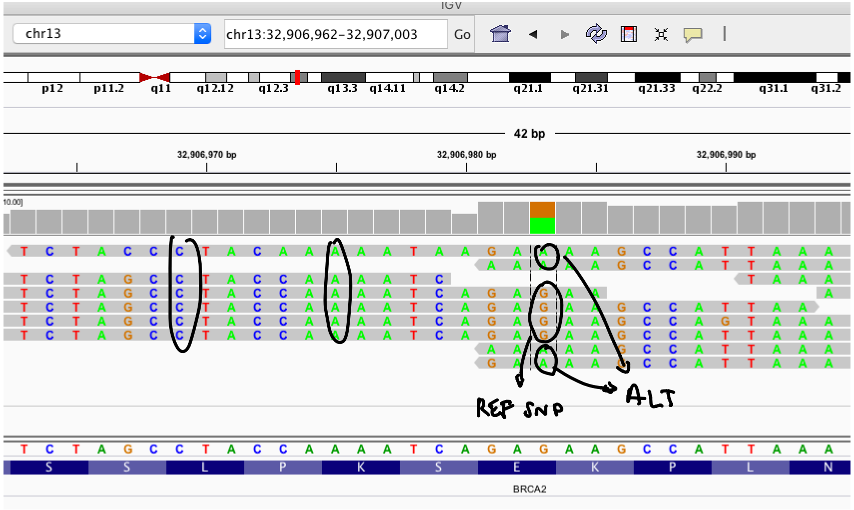
Orientation	plus
Stabilized	plus

Geno	Mag	Summary
(C;C)	1.2	one of 2 snps relevant to classifying APOE genotype

Don’t forget: humans are diploid

two . copies of each chromosome

- □ Two “copies” of each chromosome (different variants)
- □ Aligned data is a combination from both:



REF SNP

ALT

Variant data: VCF files

- Standard file format to list variants of one or many individuals
- Let’s take a look!

Variant data: VCF files

- Standard file format to list variants of one or many individuals
- Let’s take a look!
- Genotype matrix:

$$\begin{array}{cc}
 & \begin{array}{cccccc} \text{person 1} & & & & & \text{person n} \end{array} \\
\begin{array}{c} \text{SNP 1} \\ \text{SNP 2} \\ \\ \\ \\ \\ \text{SNP m} \end{array} &
\begin{bmatrix}
0|0 & 1|0 & 1|0 & 0|0 & 0|0 & 0|1 \\
0|1 & 0|0 & 1|1 & 1|0 & 0|0 & 1|1 \\
0|1 & 0|1 & 0|1 & 0|0 & 0|0 & 0|0 \\
0|0 & 0|0 & 0|0 & 1|1 & 0|1 & 1|0 \\
1|0 & 0|1 & 1|1 & 0|0 & 0|0 & 0|0 \\
0|0 & 1|1 & 0|0 & 0|0 & 0|1 & 0|0 \\
0|1 & 0|1 & 0|0 & 1|0 & 0|0 & 1|1
\end{bmatrix}
\end{array}$$

Variant data: VCF files

- Standard file format to list variants of one or many individuals
- Let's take a look!
- Genotype matrix:

$$\begin{array}{cc} & \begin{array}{cccccc}\text{person 1} & & & & & \text{person n}\end{array} \\ \begin{array}{c}\text{SNP 1}\\ \text{SNP 2}\\ \\ \\ \\ \\ \text{SNP m}\end{array} & \begin{bmatrix} 0|0 & 1|0 & 1|0 & 0|0 & 0|0 & 0|1 \\ 0|1 & 0|0 & 1|1 & 1|0 & 0|0 & 1|1 \\ 0|1 & 0|1 & 0|1 & 0|0 & 0|0 & 0|0 \\ 0|0 & 0|0 & 0|0 & 1|1 & 0|1 & 1|0 \\ 1|0 & 0|1 & 1|1 & 0|0 & 0|0 & 0|0 \\ 0|0 & 1|1 & 0|0 & 0|0 & 0|1 & 0|0 \\ 0|1 & 0|1 & 0|0 & 1|0 & 0|0 & 1|1 \end{bmatrix} \end{array} \xrightarrow[\text{count ALT occurrences}]{\text{additive effect}} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 2 & 1 & 0 & 2 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 2 \end{bmatrix}$$

Genome-Wide Association Studies (GWAS)

$$\begin{array}{r|cccccc|}
 & \text{SNP 1} & & & & & \text{SNP m} \\
\text{person 1} & 0 & 1 & 1 & 0 & 0 & 1 \\
 & 1 & 0 & 2 & 1 & 0 & 2 \\
 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 2 & 1 & 1 \\
 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 2 & 0 & 0 & 1 & 0 \\
\text{person n} & 1 & 1 & 0 & 1 & 0 & 2
\end{array}$$

Genome-Wide Association Studies (GWAS)

$$
\begin{array}{ccc}
 & \begin{array}{cccccc}\text{SNP 1} & & & & & \text{SNP m}\end{array} & \text{phenotype} \\
\begin{array}{c}\text{person 1}\\ \\ \\ \\ \\ \\ \text{person n}\end{array} &
\underbrace{\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 2 & 1 & 0 & 2 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 2
\end{bmatrix}}_{\text{genome-wide}} &
\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}
\end{array}
$$

e.g. has Diabetes

(doesn't need to be binary. e.g., height)

Genome-Wide Association Studies (GWAS)

$$\begin{array}{lcccccccc}
 & & & & \text{associated} & & & & \\
 & \text{SNP 1} & & \text{SNP}_{200} & & & \text{SNP m} & & \text{phenotype} \\
\text{person 1} & 0 & 1 & 1 & 0 & 0 & 1 & & 0 \\
 & 1 & 0 & 2 & 1 & 0 & 2 & & 0 \\
 & 1 & 1 & 1 & 0 & 0 & 0 & & 1 \\
 & 0 & 0 & 0 & 2 & 1 & 1 & & 1 \\
 & 1 & 1 & 1 & 0 & 0 & 0 & & 1 \\
 & 0 & 2 & 0 & 0 & 1 & 0 & & 0 \\
\text{person n} & 1 & 1 & 0 & 1 & 0 & 2 & & 1 \\
\text{new person} & 1 & 0 & 1 & 1 & 2 & 0 & & ?\ (\text{predict})
\end{array}$$

□ Which SNPs are associated with a given phenotype?

□ Given a new individual's genotype, can you predict their phenotype?

Revisiting Logistic Regression

□ Predict binary variable from real-valued features

$$\underset{p}{\overset{\shortparallel}{p(1 \mid \underline{x})}} = \frac{e^{\beta_0 + \underline{\beta}^T \underline{x}}}{1 + e^{\beta_0 + \underline{\beta}^T \underline{x}}} = \frac{1}{1 + e^{-(\beta_0 + \underline{\beta}^T \underline{x})}}$$

$$1 + e^{-(\beta_0 + \underline{\beta}^T \underline{x})} = \frac{1}{p} \Rightarrow e^{-(\beta_0 + \underline{\beta}^T \underline{x})} = \frac{1}{p} - 1 = \frac{1-p}{p}$$

$$\rightarrow \ln\left(\frac{p}{1-p}\right) = \underbrace{\beta_0 + \underline{\beta}^T \underline{x}}_{\text{linear model}}$$

$\frac{p}{1-p}$: odds ratio

$\ln \frac{p}{1-p}$: log odds ratio