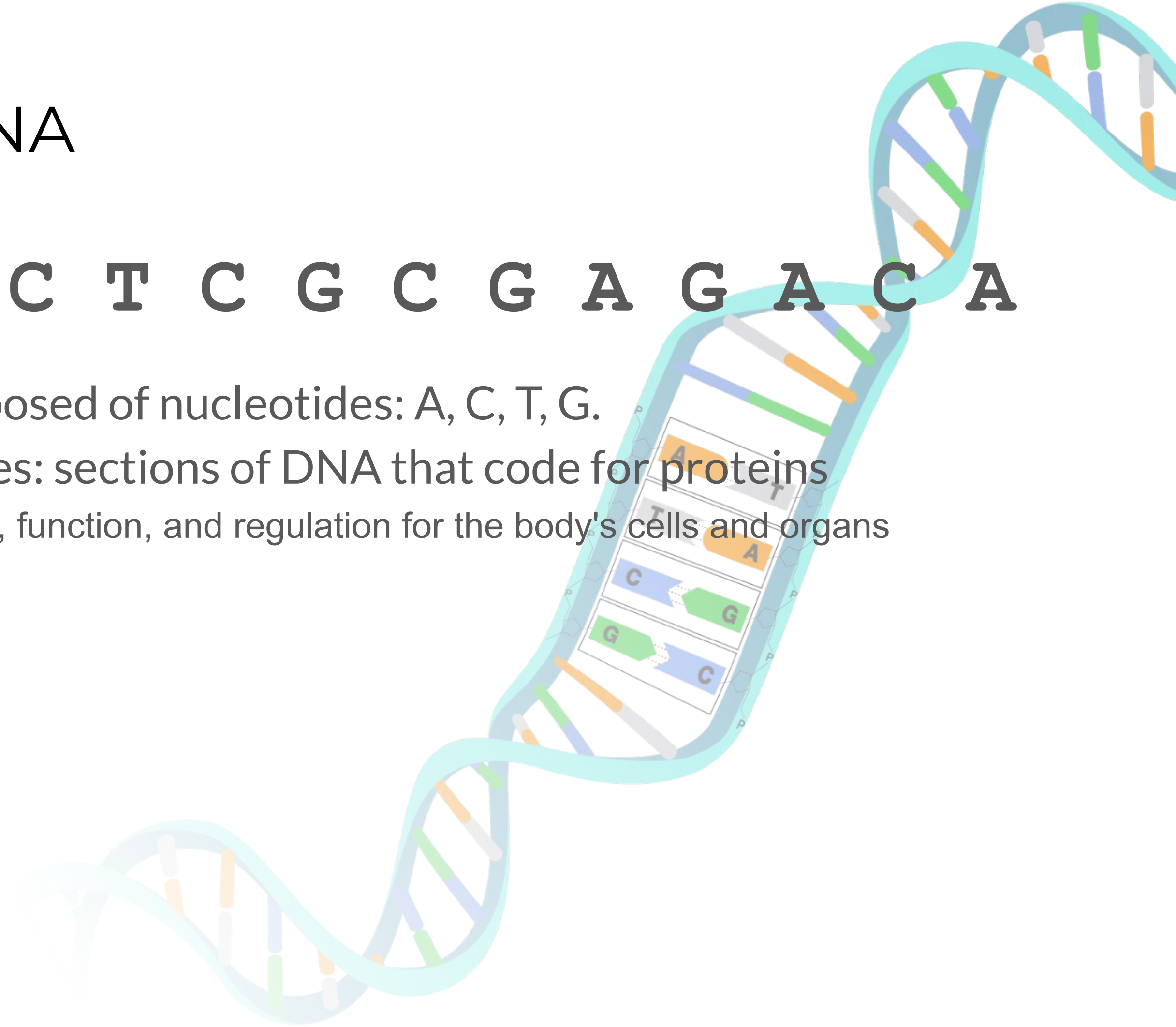
Predicting genes in DNA using a Hidden Markov Model

Iris Zhou

Background on DNA

A C T C G C G A G A C A

- DNA molecules are composed of nucleotides: A, C, T, G.
- We want to focus on genes: sections of DNA that code for proteins
  - Proteins provide structure, function, and regulation for the body's cells and organs

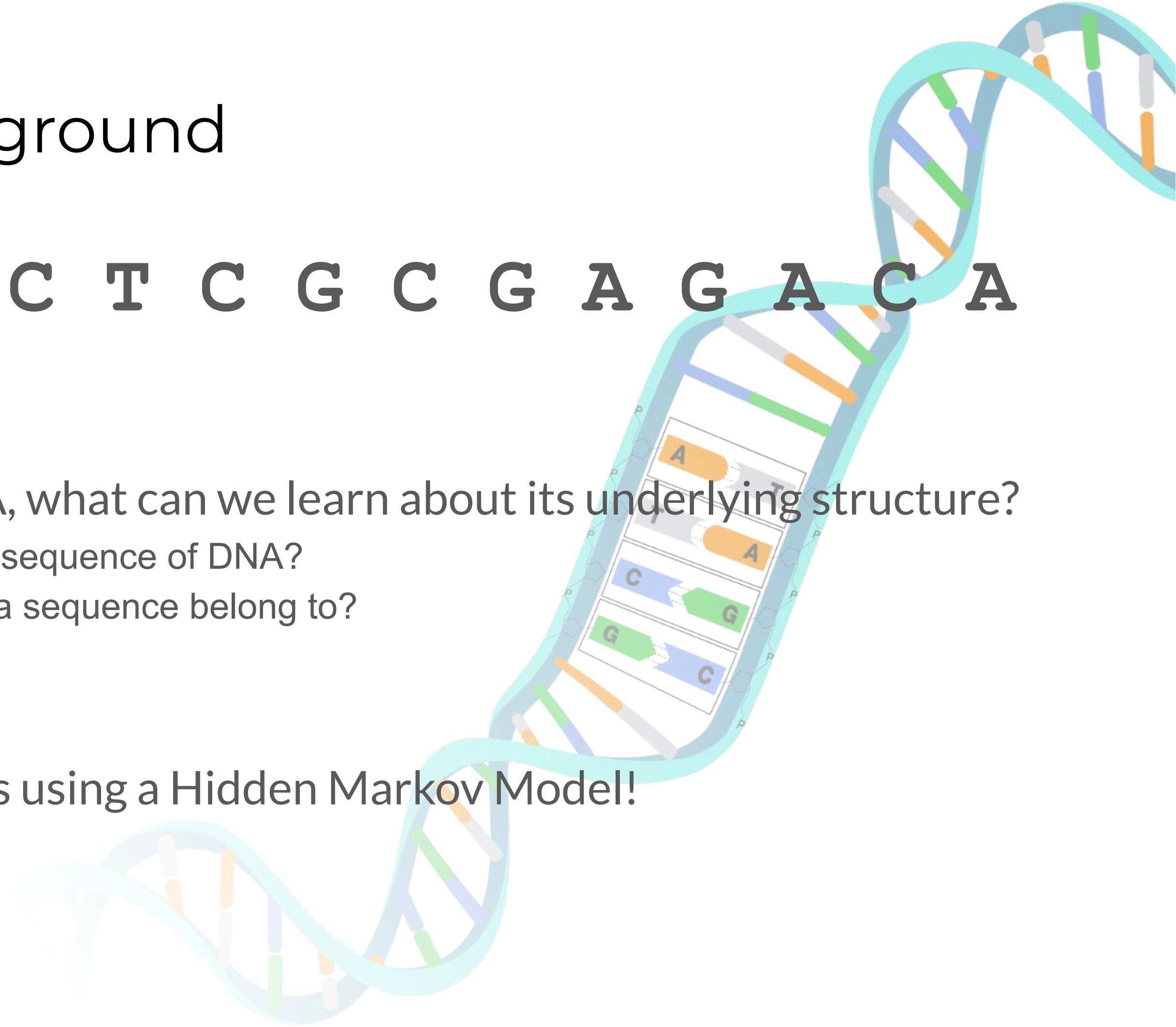


Motivation & Background

A C T C G C G A G A C A

- Given a sequence of DNA, what can we learn about its underlying structure?
  - Where are the genes in a sequence of DNA?
  - What protein family does a sequence belong to?

- We can learn these things using a Hidden Markov Model!

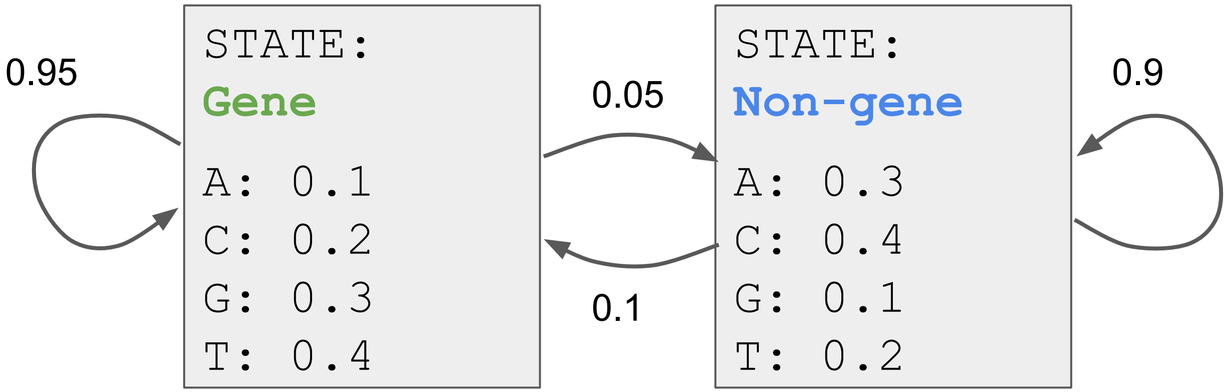


What is a Hidden Markov Model?

- Inputs: sequence & states (for each observation)
  - Outcomes of die rolls & fair or loaded die   1 3 6 2 4 2 3 1 2
  - Sequence of nucleotides & gene or non-gene   A T C G A T A G
- Outputs: transition & emission probabilities

How can we use HMMs for DNA?

Sequence:	**A**	**C**	**T**	**C**	**G**	**C**	**G**	**A**	**G**	**A**	**C**	**A**
Hidden States:	**G**	**G**	**G**	**N**	**N**	**N**	**G**	**G**	**G**	**G**	**N**	**N**



0.95

STATE:
Gene

A: 0.1
C: 0.2
G: 0.3
T: 0.4

0.05

0.1

STATE:
Non-gene

A: 0.3
C: 0.4
G: 0.1
T: 0.2

0.9

emission probabilities

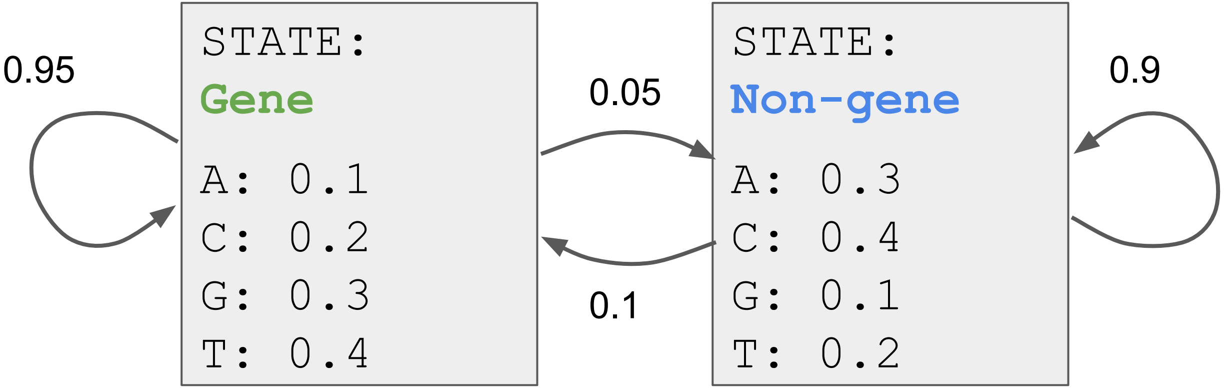
transition probabilities

emission probabilities

	A	C	G	T
G	0.1	0.2	0.3	0.4
N	0.3	0.4	0.1	0.2

transition probabilities

	G	N
G	0.95	0.05
N	0.1	0.9



STATE:
Gene

A: 0.1
C: 0.2
G: 0.3
T: 0.4

STATE:
Non-gene

A: 0.3
C: 0.4
G: 0.1
T: 0.2

Viterbi Algorithm

- **Goal:** For each symbol in the sequence, what's the most probable sequence of states that ends in that symbol?

Ex:

Possible sequences of states:
(ending in T)

A	C	T
G	G	G
G	N	G
N	G	G
N	N	G

. . .

Viterbi Algorithm

- **Goal:** For each symbol in the sequence, what's the most probable sequence of states that ends in that symbol?
- **Inputs:** a sequence, an emissions matrix, and a transitions matrix

```
Sequence:
A  C  T  G...
```

emission probabilities

	A	C	G	T
G	0.1	0.2	0.3	0.4
N	0.3	0.4	0.1	0.2

transition probabilities

	G	N
G	0.95	0.05
N	0.1	0.9

Viterbi Algorithm

- **Goal:** For each symbol in the sequence, what's the most probable sequence of states that ends in that symbol?
- **Inputs:** a sequence, an emissions matrix, and a transitions matrix
- **Outputs:** the hidden state sequence

Ex:

Possible sequences of states:
(ending in T)

A	C	T	P(seq)
G	G	G	0.102
G	N	G	0.042
N	G	G	0.057
N	N	G	0.092
...			

Viterbi Algorithm

- **Goal:** For each symbol in the sequence, what's the most probable sequence of states that ends in that symbol?
- **Inputs:** a sequence, an emissions matrix, and a transitions matrix
- **Outputs:** the hidden state sequence
- **Limitations:** Algorithm assumes that the emission and transition probabilities are already known
    - Estimate by counting symbols and transitions between symbols with known genes

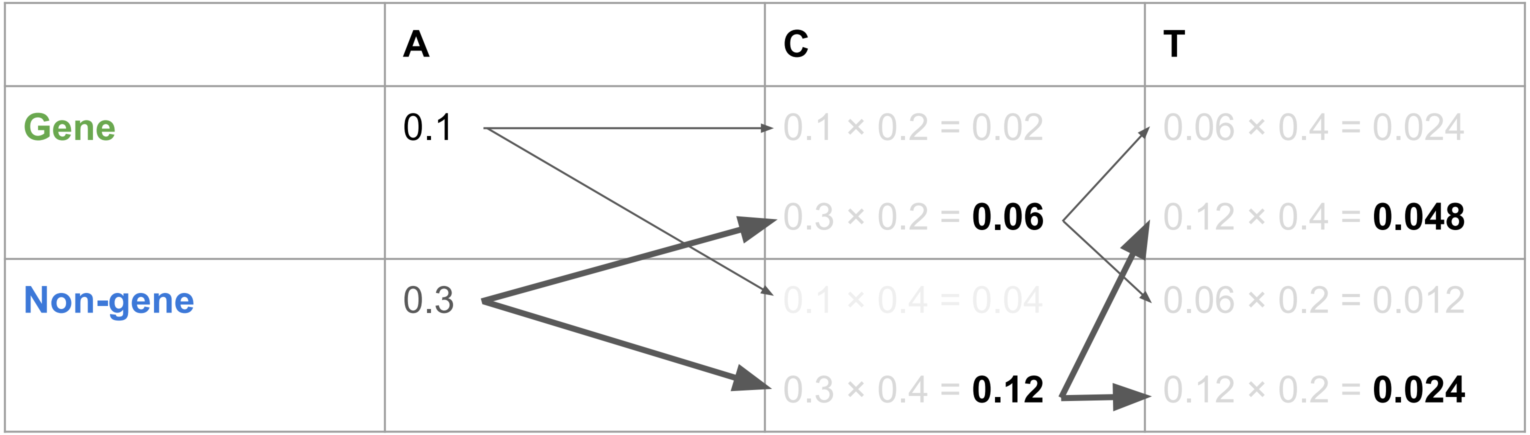
```
Sequence:
A  C  T
```

emission probabilities

	A	C	G	T
G	0.1	0.2	0.3	0.4
N	0.3	0.4	0.1	0.2

transition probabilities

	G	N
G	0.95	0.05
N	0.1	0.9



	A	C	T
Gene	0.1	0.1 × 0.2 = 0.02 0.3 × 0.2 = **0.06**	0.06 × 0.4 = 0.024 0.12 × 0.4 = **0.048**
Non-gene	0.3	0.1 × 0.4 = 0.04 0.3 × 0.4 = **0.12**	0.06 × 0.2 = 0.012 0.12 × 0.2 = **0.024**

Sequence:

A C T

N N G

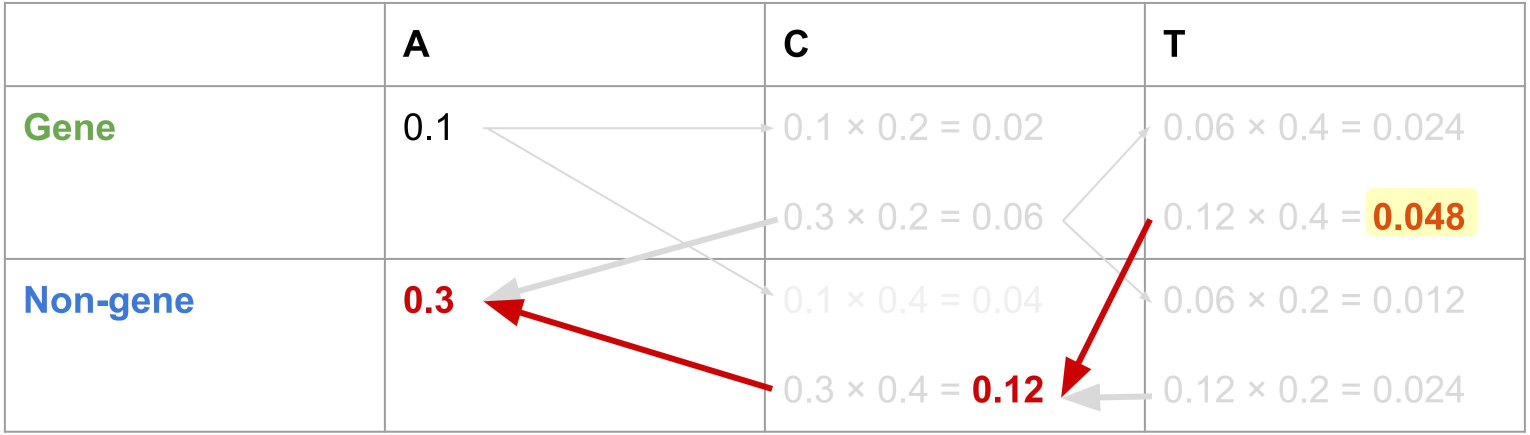
emission probabilities

	A	C	G	T
G	0.1	0.2	0.3	0.4
N	0.3	0.4	0.1	0.2

transition probabilities

	G	N
G	0.95	0.05
N	0.1	0.9

⇦ traceback



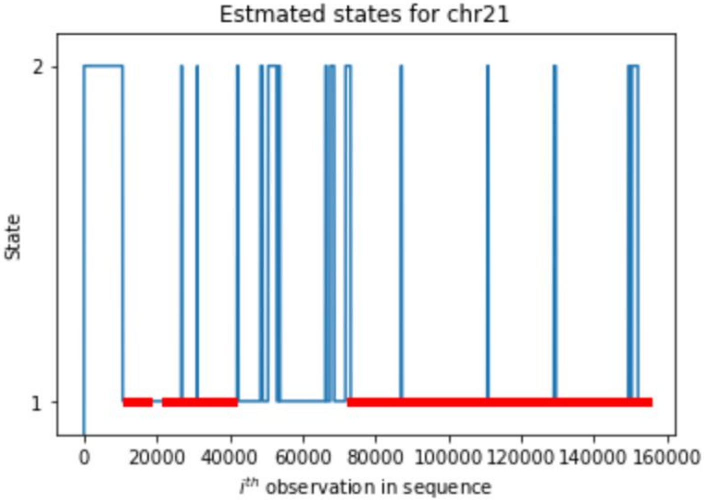
	A	C	T
Gene	0.1	0.1 × 0.2 = 0.02 0.3 × 0.2 = 0.06	0.06 × 0.4 = 0.024 0.12 × 0.4 = **0.048**
Non-gene	**0.3**	0.1 × 0.4 = 0.04 0.3 × 0.4 = **0.12**	0.06 × 0.2 = 0.012 0.12 × 0.2 = 0.024

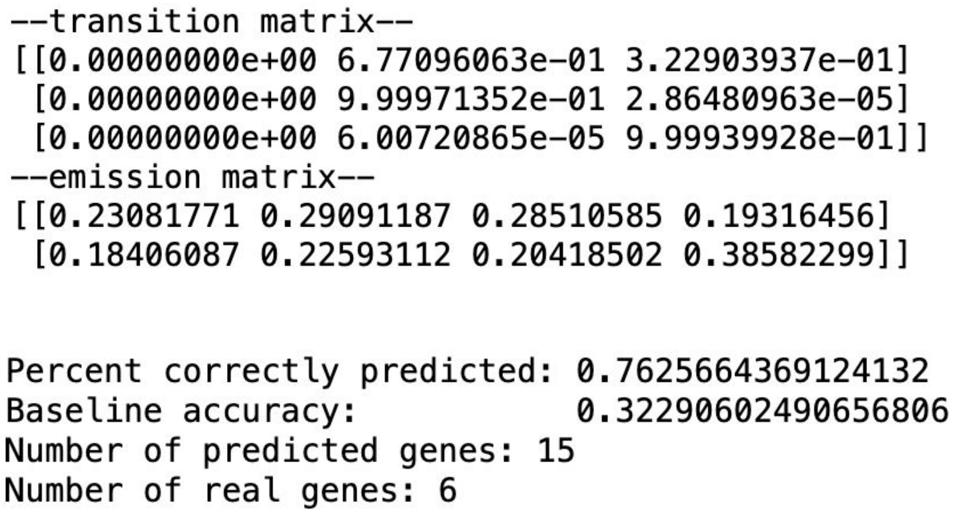
Building a model using Viterbi

- Idea:
  - Use known genes in Chromosome 21 to estimate emission and transition probabilities
  - Use Viterbi on sequences of DNA to predict locations of genes
  - Compare accuracy of predictions to locations of genes

Results

[5000000, 5154658]



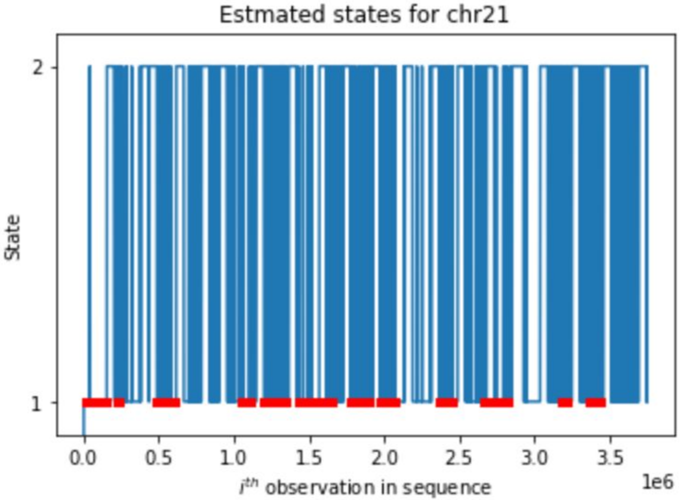


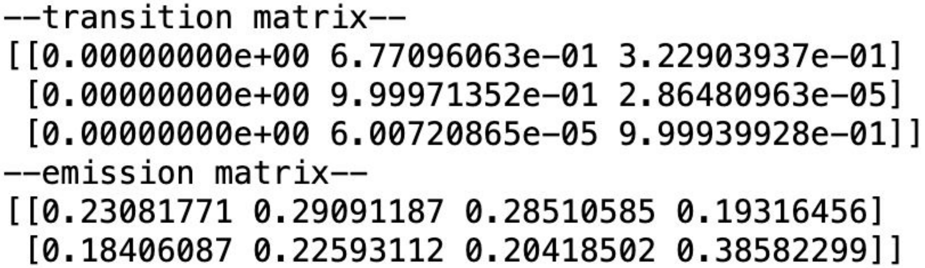
```
--transition matrix--
[[0.00000000e+00 6.77096063e-01 3.22903937e-01]
 [0.00000000e+00 9.99971352e-01 2.86480963e-05]
 [0.00000000e+00 6.00720865e-05 9.99939928e-01]]
--emission matrix--
[[0.23081771 0.29091187 0.28510585 0.19316456]
 [0.18406087 0.22593112 0.20418502 0.38582299]]


Percent correctly predicted: 0.762566436912413 2
Baseline accuracy:           0.32290602490656806
Number of predicted genes: 15
Number of real genes: 6
```

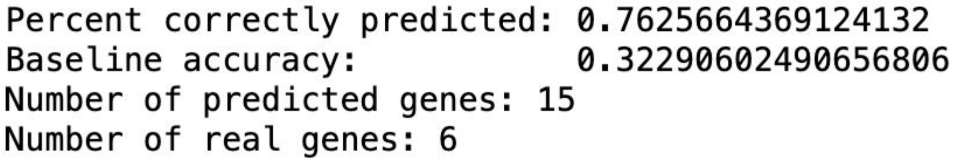
Results

```
[5011799, 8761335]
```





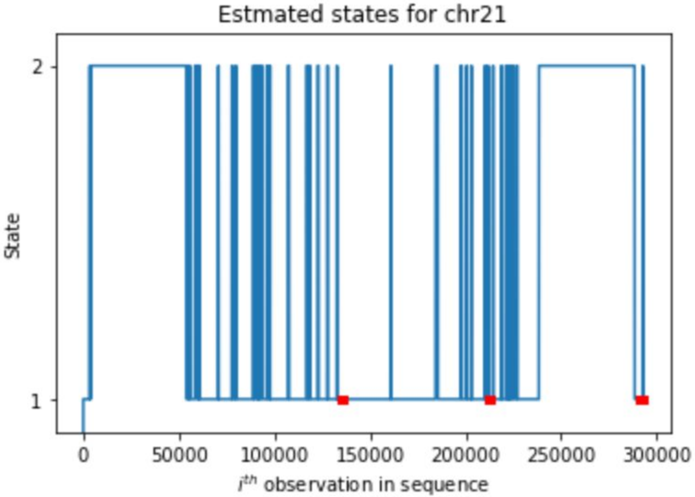
```
--transition matrix--
[[0.00000000e+00 6.77096063e-01 3.22903937e-01]
 [0.00000000e+00 9.99971352e-01 2.86480963e-05]
 [0.00000000e+00 6.00720865e-05 9.99939928e-01]]
--emission matrix--
[[0.23081771 0.29091187 0.28510585 0.19316456]
 [0.18406087 0.22593112 0.20418502 0.38582299]]
```

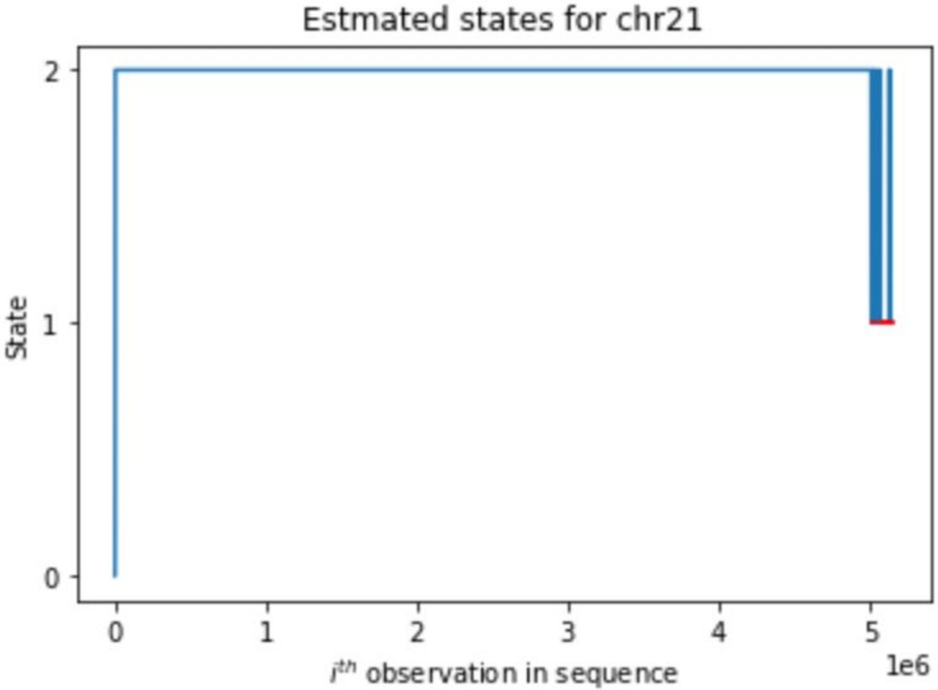


```
Percent correctly predicted: 0.7625664369124132
Baseline accuracy:           0.32290602490656806
Number of predicted genes: 15
Number of real genes: 6
```

Results

```
[5011799, 8761335]
```





Limitations and Improvements for Model

- Optimize model performance and algorithm efficiency
- Model may be too naive
  - "Gene" or "non-gene" status is not directly dependent on single nucleotides
  - Use codons (groups of 3 nucleotides) instead of single nucleotides
- Model depends on already knowing probabilities

Overall Takeaways

- Hidden Markov Models can help us model hidden states in a sequence
    - Gene vs. non-gene
    - Fair vs. loaded die
    - Speech recognition (what sound is being emitted?)
- Finding hidden states can help us better understand which sections of DNA are important and discover where underlying processes are occuring

Thank you!