Markov chains, part 2

Ben Langmead

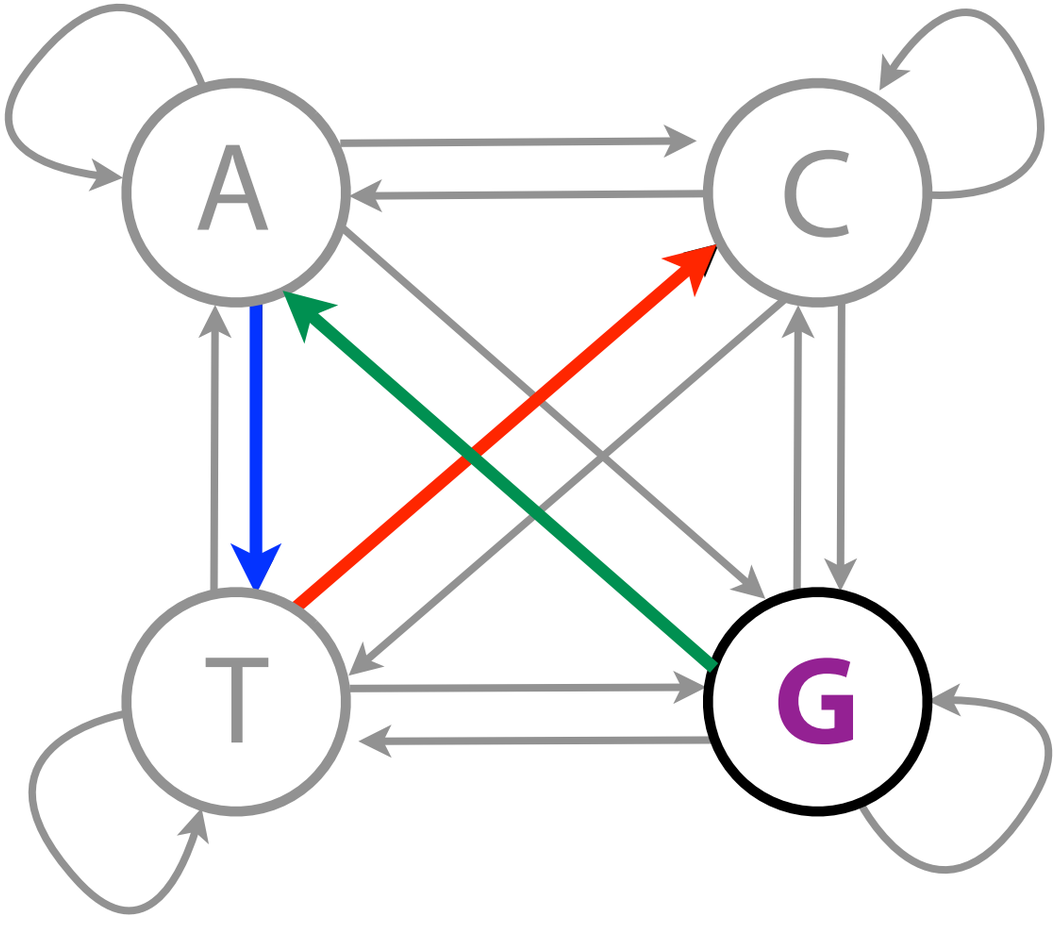
Markov chain

```
>>> iTab, oTab = islandTransitionTables(fn, ifn)
>>> print(iTab)
[[ 0.18427153, 0.27129525, 0.4055757 , 0.13885752],
 [ 0.19081672, 0.36113346, 0.24897947, 0.19907035],
 [ 0.17440554, 0.32764433, 0.35676759, 0.14118254],
 [ 0.09348595, 0.3474561 , 0.36885   , 0.19020795]]
```

X_{i-1} A C G T

A C G T

X_i



$x = \text{GATC}$

$P(x) = P(x_4 \mid x_3)\, P(x_3 \mid x_2)\, P(x_2 \mid x_1)\, P(x_1)$

$P(x) = P(\text{C} \mid \text{T})\, P(\text{T} \mid \text{A})\, P(\text{A} \mid \text{G})\, P(\text{G})$

$= 0.347 * 0.139 * 0.174 * 0.25$

$= 0.0021$

Markov chain

** is exponentiation

```
>>> 0.5 ** 1
0.5
>>> 0.5 ** 10
0.0009765625
>>> 0.5 ** 100
7.888609052210118e-31
>>> 0.5 ** 1000
9.33263618503218 9e-302
>>> 0.5 ** 1100
```

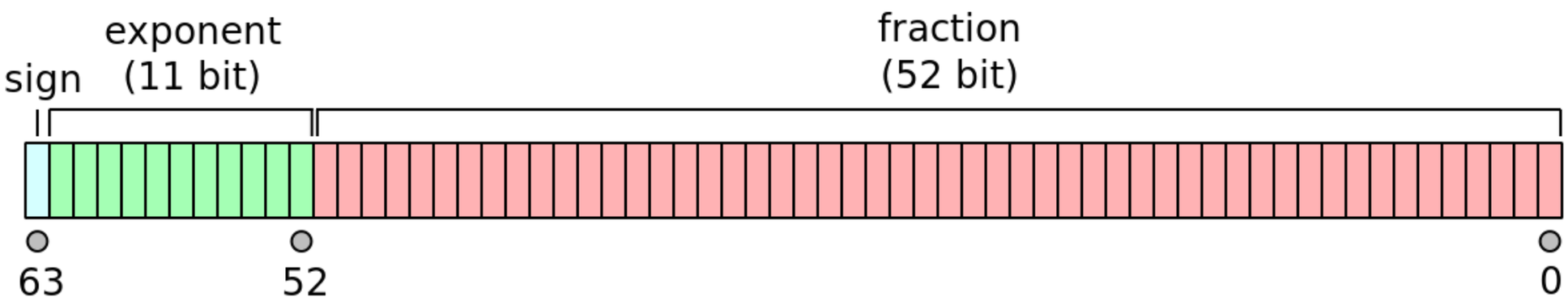
Markov chain

Underflow: when result is a number so small, it cannot be represented in a limited-precision floating point number and is **rounded to 0**

** is exponentiation

```
>>> 0.5 ** 1
0.5
>>> 0.5 ** 10
0.0009765625
>>> 0.5 ** 100
7.888609052210118e-31
>>> 0.5 ** 1000
9.33263618503218 9e-302
>>> 0.5 ** 1100
0.0
```

😱😱😱



https://en.wikipedia.org/wiki/Double-precision_floating-point_format

Markov chain

To avoid underflow, switch to log domain

$$\log P(x) \approx \log [\, P(x_k \mid x_{k-1})\, P(x_{k-1} \mid x_{k-2}) \ldots P(x_2 \mid x_1)\, P(x_1)\,]$$

$$= \log P(x_k \mid x_{k-1}) + \log P(x_{k-1} \mid x_{k-2}) + \ldots$$

…product becomes sum!

$$= \sum_{i=2}^{k} \log P(x_i \mid x_{i-1}) + \log P(x_1)$$

Assume logs are base 2

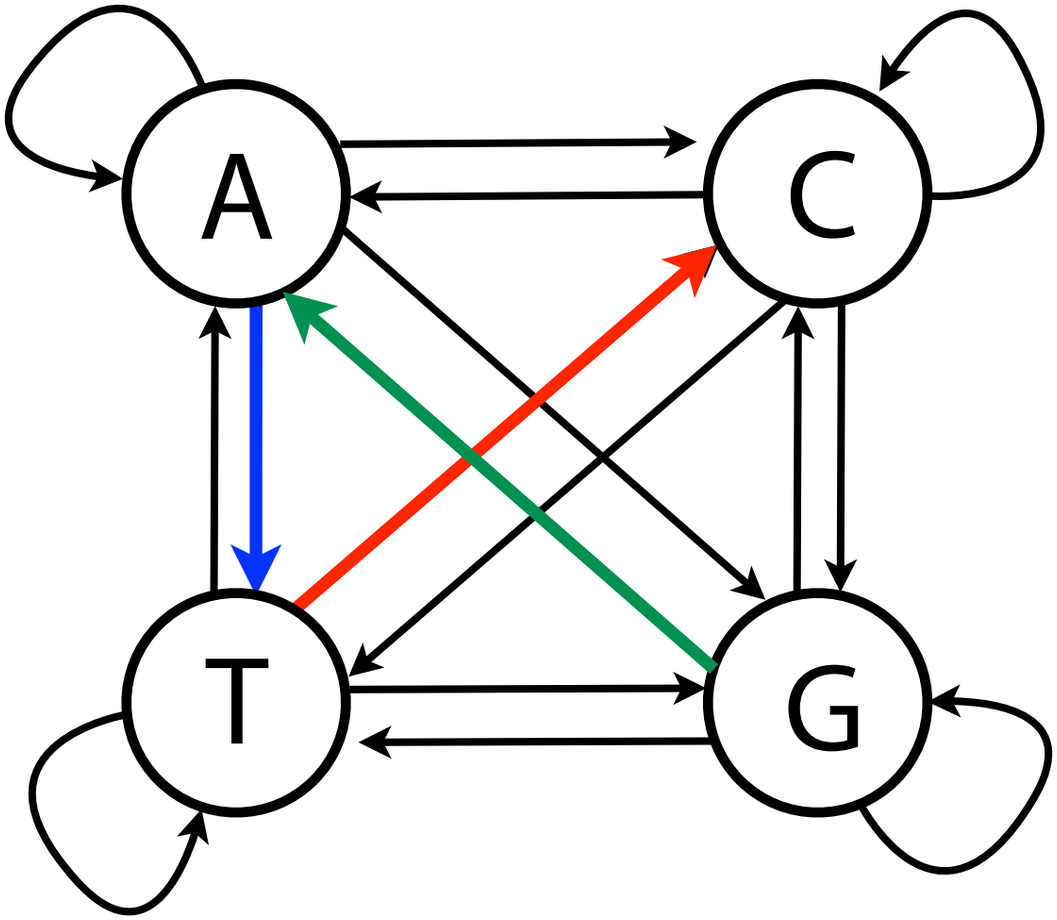
Markov chain

```
>>> iTab, nTab = islandTransitionTables(fn, ifn)
>>> print(numpy.log2(iTab))
[[-2.44009488, -1.8820643 , -1.30195688, -2.84832282],
 [-2.38974049, -1.469396  , -2.00590131, -2.32864974],
 [-2.51948223, -1.60979755, -1.48694353, -2.82436637],
 [-3.41910668, -1.52509737, -1.43889385, -2.39435058]]
```

X_{i-1}: **A** **C** **G** **T**

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

X_i



$x = \text{GATC}$

$$\log P(x) = \sum_{i=2}^{4} \log P(x_i \mid x_{i-1}) + \log P(x_1)$$

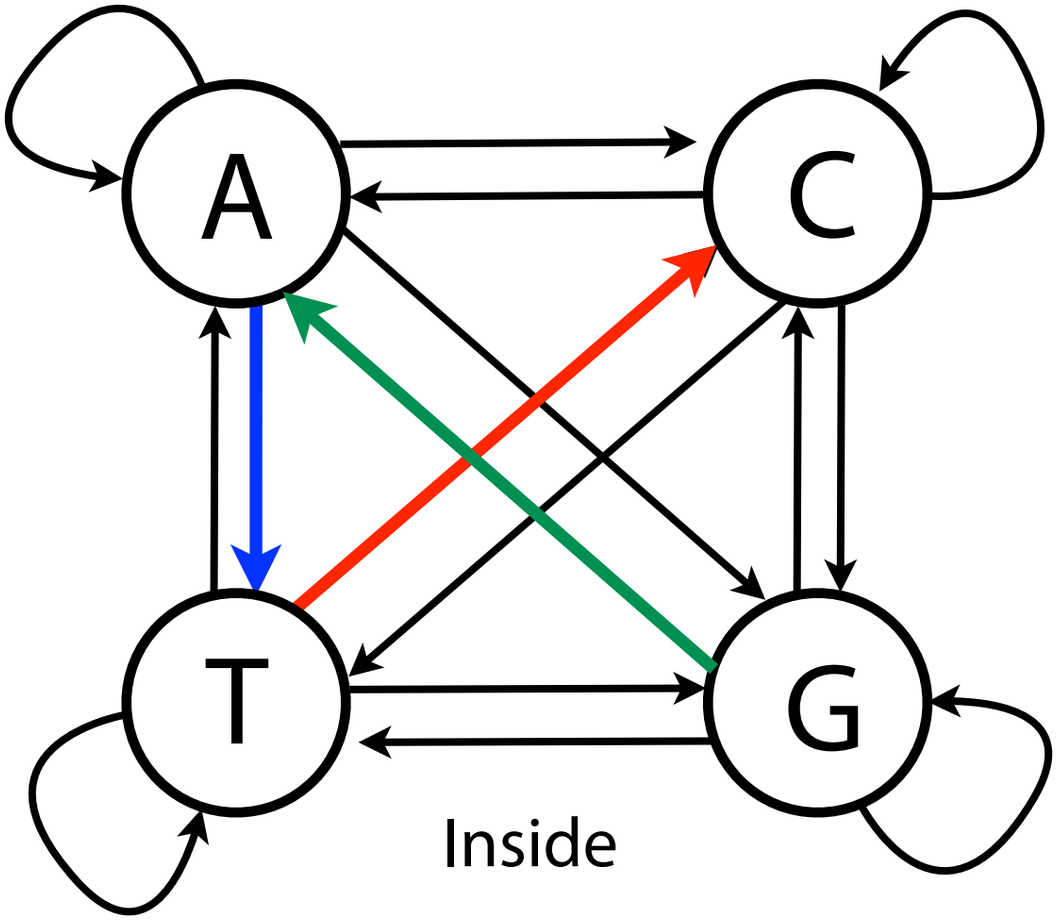
$$= -1.525 + -2.848 + -2.519 + -2.0$$

$$= -8.892$$

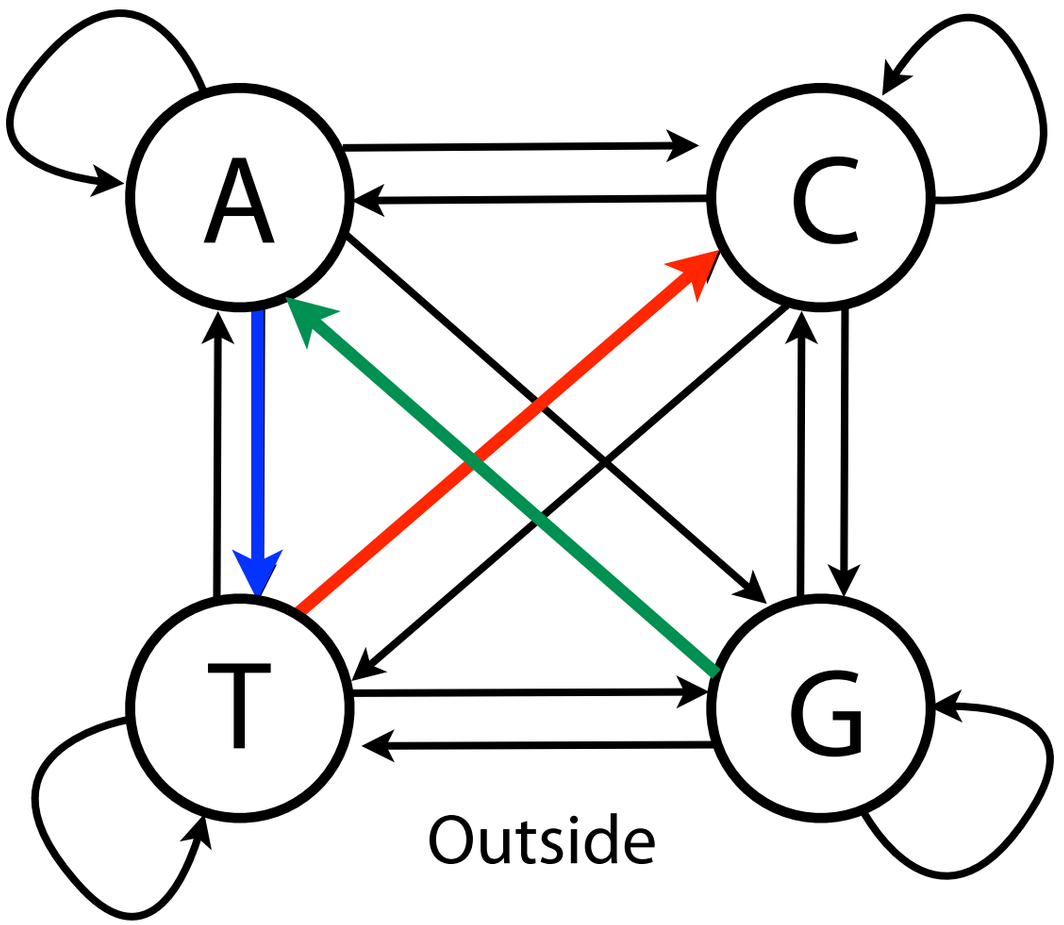
Markov chain

P(x) given the inside-CpG model is helpful, but we really want to know which model is better, inside CpG or outside CpG?

Use *ratio:* $$\frac{\text{P(} x \text{) using model on the left}}{\text{P(} x \text{) using model on the right}}$$



Inside



Outside

Markov chain

Take log, get a *log ratio*: $$S(x) = \log \frac{P(x) \text{ inside CpG}}{P(x) \text{ outside CpG}}$$

If inside more probable than outside, fraction is > 1, log ratio is > 0.
Otherwise, fraction is ≤ 1 and log ratio is ≤ 0.

$$S(x) = \log \frac{P(x) \text{ inside CpG}}{P(x) \text{ outside CpG}}$$

$$= \log [\, P(x) \text{ inside CpG} \,] - \log [\, P(x) \text{ outside CpG} \,]$$

(Marginal probabilities ignored here)

$$= \sum_{i=2}^{k} \left(\log [\, P(x_i \mid x_{i-1}) \text{ inside CpG} \,] \right) - \sum_{i=2}^{k} \log \left([\, P(x_i \mid x_{i-1}) \text{ outside CpG} \,] \right)$$

$$= \sum_{i=2}^{k} \underbrace{\left(\log [\, P(x_i \mid x_{i-1}) \text{ inside CpG} \,] - \log [\, P(x_i \mid x_{i-1}) \text{ outside CpG} \,] \right)}$$

New table: elementwise log ratios between inside/outside

Markov chain

```
>>> iTab, oTab = islandTransitionTables(fn, ifn)
>>> print(iTab)
[[ 0.18427153, 0.27129525, 0.4055757 , 0.13885752],
 [ 0.19081672, 0.36113346, 0.24897947, 0.19907035],
 [ 0.17440554, 0.32764433, 0.35676759, 0.14118254],
 [ 0.09348595, 0.3474561 , 0.36885   , 0.19020795]]
>>> print(oTab)
 [[0.33804066, 0.17971034, 0.23104207, 0.25120694],
  [0.37777025, 0.25612117, 0.03987225, 0.32623633],
  [0.30257815, 0.20326794, 0.24910719, 0.24504672],
  [0.21790184, 0.20942905, 0.2642385 , 0.3084306 ]]
>>> combinedTab = numpy.log2(iTab) - numpy.log2(oTab)
>>> print(combinedTab)
 [[-0.87536356,  0.59419041,  0.81181564, -0.85527103],
  [-0.98532149,  0.49570561,  2.64256972, -0.7126391 ],
  [-0.79486196,  0.68874785,  0.51821792, -0.79549511],
  [-1.22085697,  0.73036913,  0.48119354, -0.69736839]]
```

Inside: A C G T

Outside: A C G T

Log ratio: A C G T

A C G T

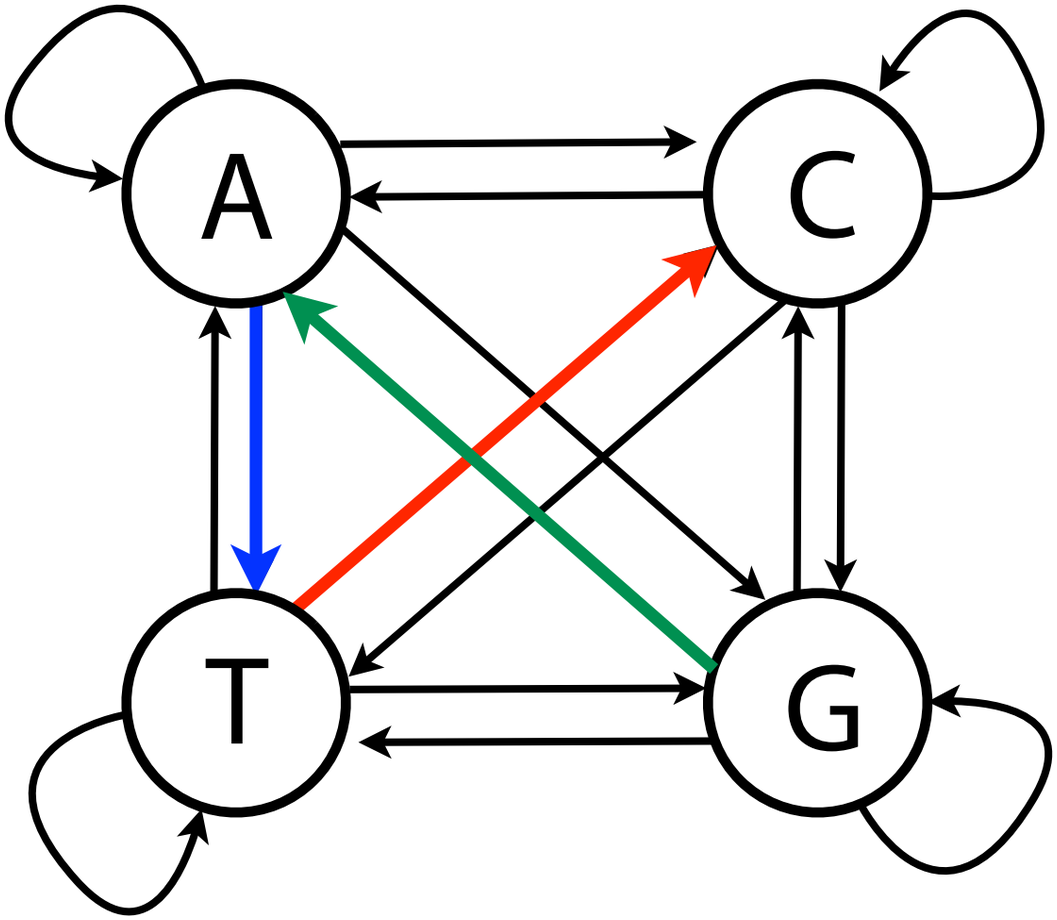
Markov chain

```
>>> iTab, oTab = islandTransitionTables(fn, ifn)
>>> combinedTab = numpy.log2(iTab) - numpy.log2(oTab)
>>> print(combinedTab)
[[-0.87536356,  0.59419041,  0.81181564, -0.85527103],
 [-0.98532149,  0.49570561,  2.64256972, -0.7126391 ],
 [-0.79486196,  0.68874785,  0.51821792, -0.79549511],
 [-1.22085697,  0.73036913,  0.48119354, -0.69736839]]
```

X_{i-1} A C G T

A C G T

X_i



(Marginal probabilities ignored here)

x = GATC

S(x) = 0.730 +
-0.855 +
-0.795

= -0.92

Negative, therefore probability with *outside* model is greater

Markov chain

$$S(x) = \log \frac{P(x) \text{ inside CpG}}{P(x) \text{ outside CpG}}$$

S(CGCGCGCGCGCGCGCGCGCGCGCG) = 32.246609048

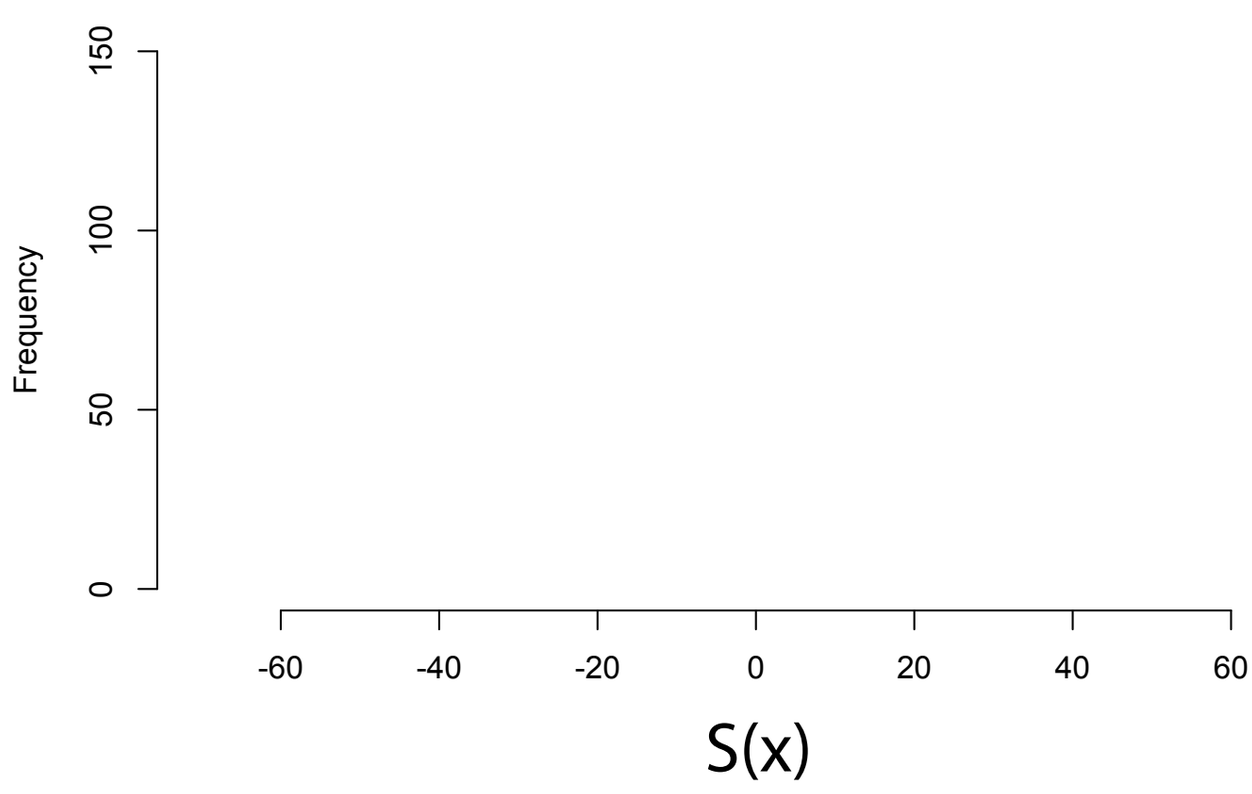
S(ATTCTACTATCATCTATCTATCTTCT) = -9.501209765

Markov chain

How well does S(x) distinguish strings drawn from inside versus outside CpG islands?

Draw 1,000 100-mers from inside CpG islands on chromosome 1, and another 1,000 from outside, calculate log ratios for all

Markov chain trained on chromosome 22 islands



Markov chain

How well does S(x) distinguish strings drawn from inside versus outside CpG islands?

Draw 1,000 100-mers from inside CpG islands on chromosome 1, and another 1,000 from outside, calculate log ratios for all

Markov chain trained on chromosome 22 islands