String Algorithms and Data Structures
Markov Chains

CS 199-225
Brad Solomon

November 29, 2022

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
Department of Computer Science

Learning Objectives

Introduce Markov Chains

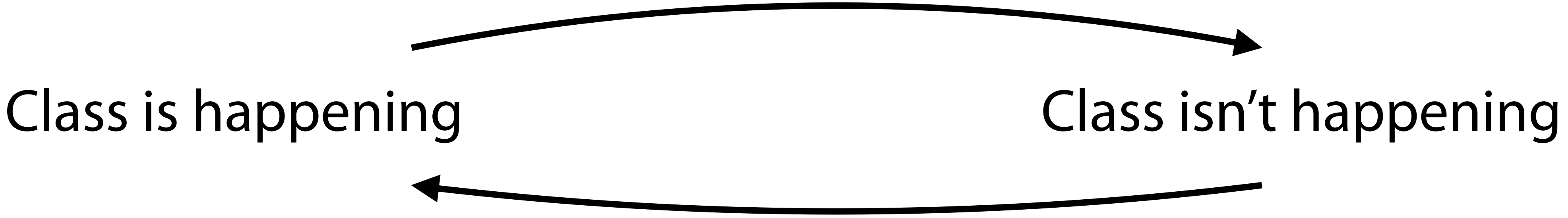
Define and determine stationary states

Identify common Markov Chain irregularities

Introduce Hidden Markov Models

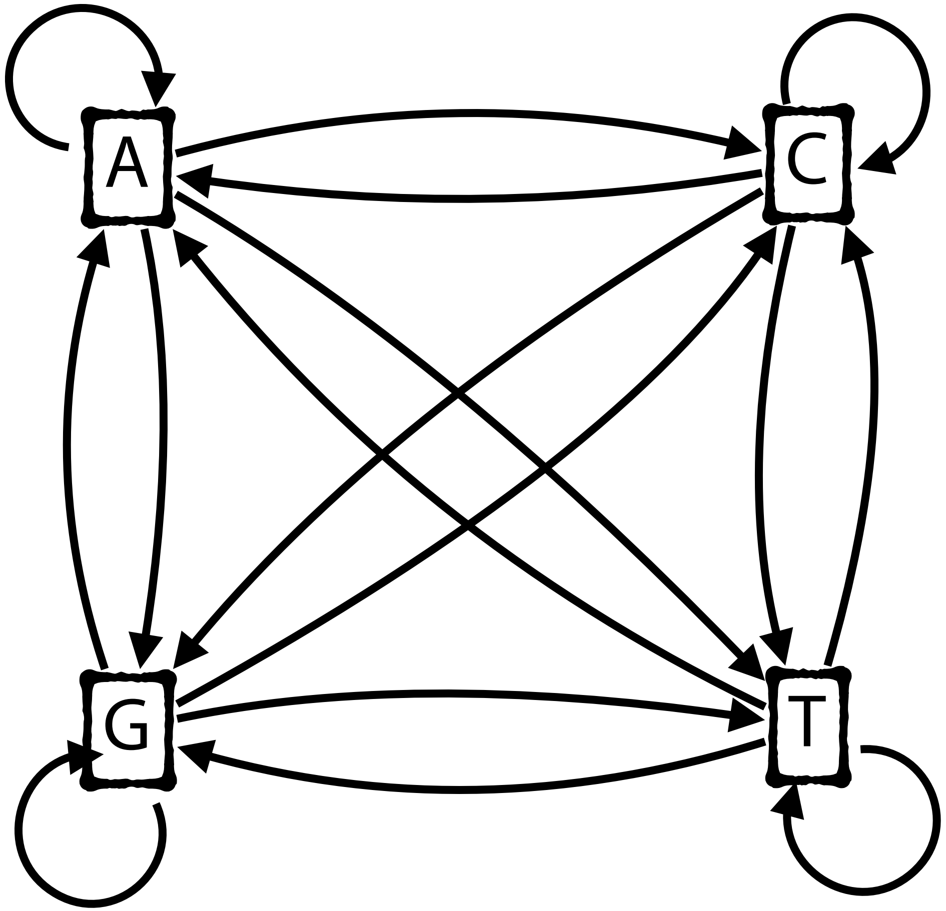
Modeling events with State Diagrams

A **state diagram** is a (usually weighted) directed graph where nodes are states and edges are transitions between them



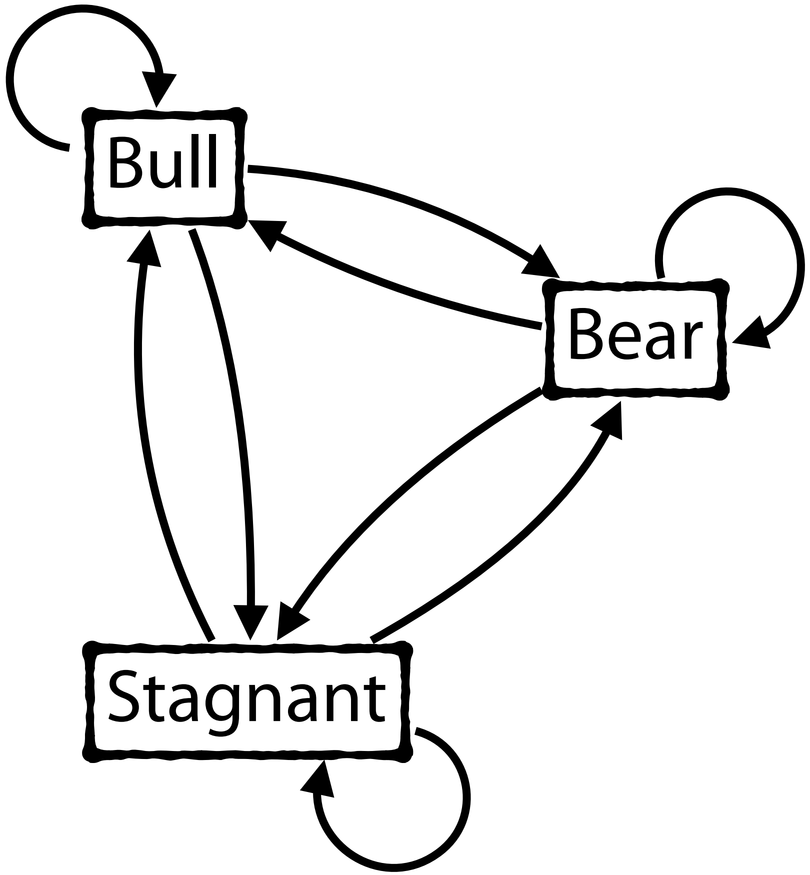
These diagrams are very useful in modeling many real world scenarios!

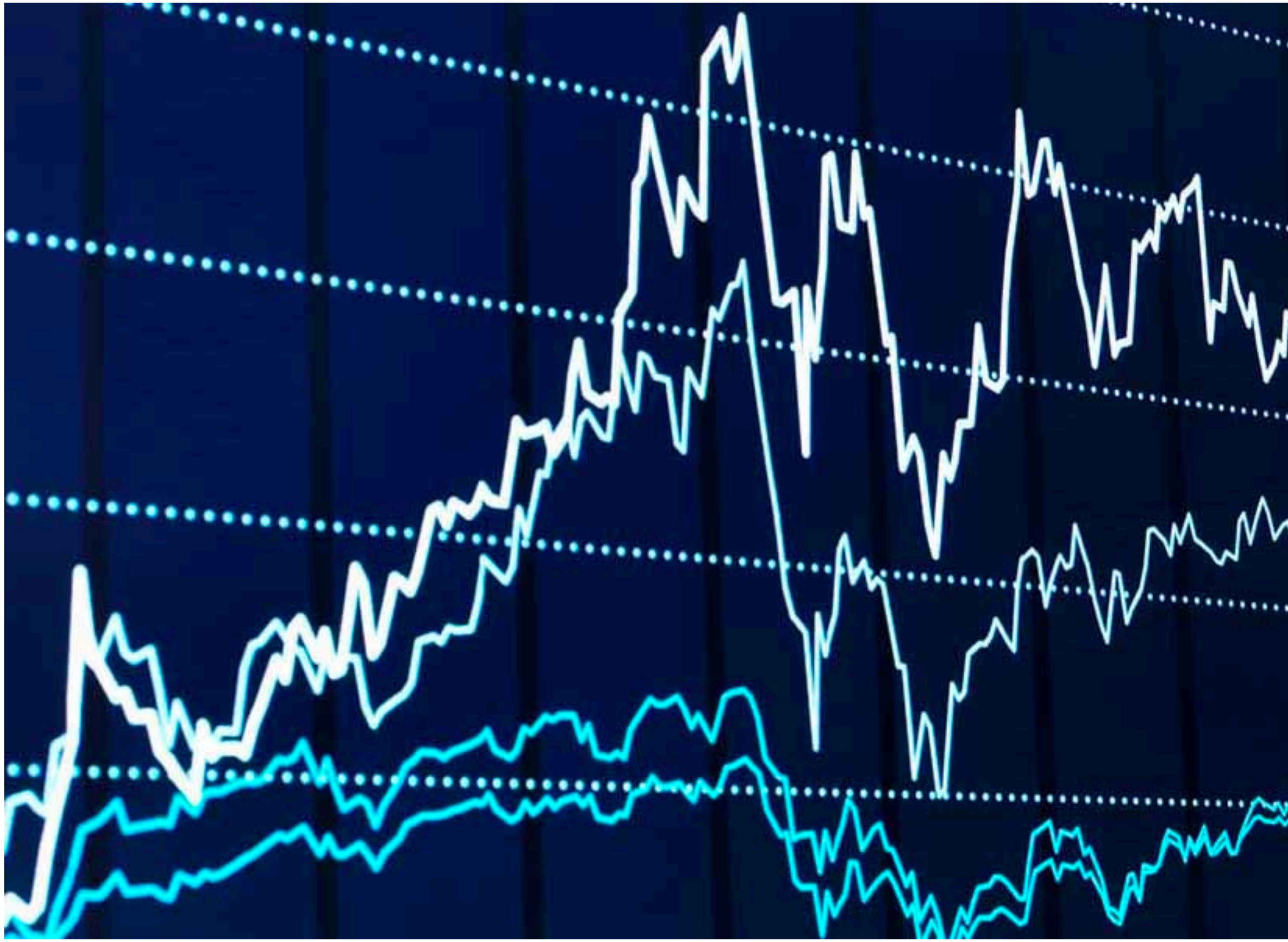
Sequence Modeling in Biology



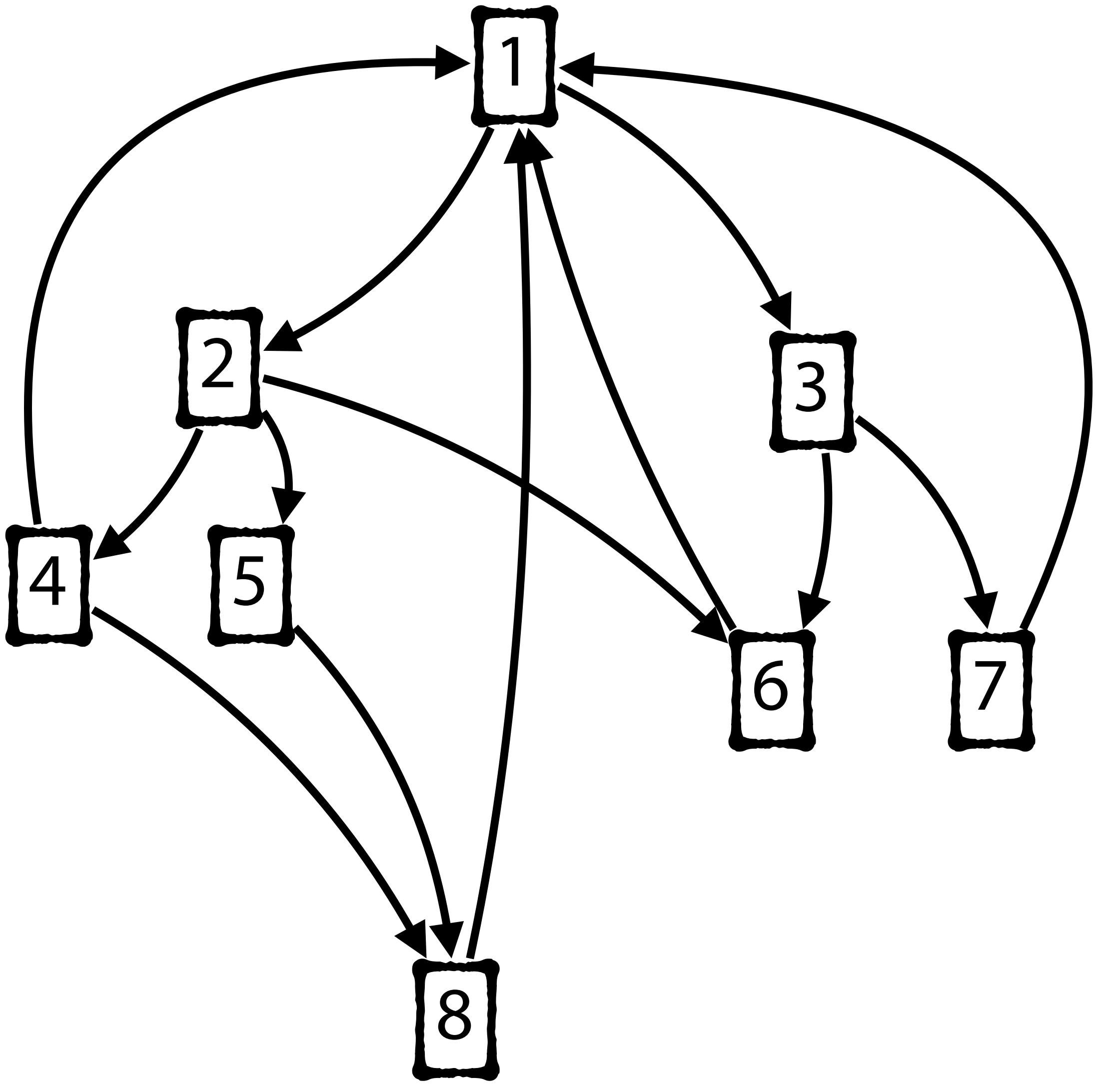
CATGACGTCGCGGACAACCCAGAATTGTCTTGAGCGATGGTAAGATCTAACCTCACTGC
CTGGGGCTTTACTGATGTCATACCGTCTTGCACGGGGATAGAATGACGGTGCCCGTGTC
ATTTTCTGAAAGTTACAGACTTCGATTAAAAAGATCGGACTGCGCGTGGGCCCGGAGAG
TTTTTCGACGTGTCAAGGACTCAAGGGAATAGTTTGGCGGGAGCGTTACAGCTTCAATT
CGATAAAATTCAACTACTGGTTTCGGCCTAATAGGTCACGTTTTATGTGAAATAGAGGG
CCCTGGGTGTTCTATGATAAGTCCTGCTTTATAACACGGGGCGGTTAGGTTAAATGACT
ATCCAAGCGCCCGCTAATTCTGTTCTGTTAATGTTCATACCAATACTCACATCACATTA
AGCCCAGTCGCAAGGGTCTGCTGCTGTTGTCGACGCCTCATGTTACTCCTGGAATCTAC
GGTTAAGGCGTGTGATCGACGATGCAGGTATACATCGGCTCGGACCTACAGTGGTCGAT
TCGCGGTTCGGCGCGTAGTTGAGTGCGATAACCCAACCGGTGGCAAGTAGCAAGAAGAC
AGACAACCTAACTAATAGTCTCTAACGGGGAATTACCTTTACCAGTCTCATGCCTCCAA
CAATGATATCGCCCACAGAAAGTAGGGTCTCAGGTATCGCATACGCCGCGCCCGGGTCC
GACAGTAGAGAGCTATTGTGTAATTCAGGCTCAGCATTCATCGACCTTTCCTGTTGTGA
TCTCGTCCGTAACGATCTGGGGGGCAAAACCGAATATCCGTATTCTCGTCCTACGGGTC
TGCGCGTGATCGTCAGTTAAGTTAAATTAATTCAGGCTACGGTAAACTTGTAGTGAGCT
ACGGGTTCGCTACAGATGAACTGAATTTATACACGGACAACTCATCGCCCATTTGGGCG
AAAGTGGCAGATTAGGAGTGCTTGATCAGGTTAGCAGGTGGACTGTATCCAACAGCGCA
CCAAAGCGTTGTAGTGGTCTAAGCACCCCTGAACAGTGGCGCCCATCGTTAGCGTAGTA
AGGTGCGACATGGGGCCAGTTAGCCTGCCCTATATCCCTTGCACACGTTCAATAAGAGG
TTTTTAAATTAGGATGCCGACCCCATCATTGGTAACTGTATGTTCATAGATATTTCTTC
AGCTGACACGCAAGGGTCAACAATAATTTCTACTATCACCCCGCTGAACGACTGTCTTT
CTTAGATTCGCGTCCTAACGTAGTGAGGGCCGAGTCATATCATAGATCAGGCATGAGAA
CACACGAGTTGTAAACAACTTGATTGCTATACTGTAGCTACCGCAAGGATCTCCTACAT
ATCTGGATCCGAGTCAGAAATACGAGTTAATGCAAATTTACGTAGACCGGTGAAAACAC
AGACCGTAGTCAGAAGTGTGGCGCGCTATTCGTACCGAACCGGTGGAGTATACAGAATT
AGGAGCTCGGTCCCCAATGCACGCCAAAAAAGGAATAAAGTATTCAAACTGCGCATGGT
CTATTATCCATCCGAACGTTGAACCTACTTCCTCGGCTTATGCTGTCCTCAACAGTATC
CGGCTGTGGATCTTAACGGCCACATTCTTAATTCCGACCGATCACCGATCGCCTTTCCT
ACTAAGTTATCCAGATCAAGGTTTGAACGGACTCGTATGACATGTGTGACTGAACCCGG
CTGTTTCAAGGCCTCTGCTTTGGTATCACTCAATATATTCAGACCAGACAAGTGGCAAA
CTAGGTATTCACGCAACCGTCGTAACATGCACTAAGGATAACTAGCGCCAGGGGGGCAT
AAAGACTACCCTATGGATTCCTTGGAGCGGGGACAATGCAGACCGGTTACGACACAATT
GGTATTATTAGCAAGACAATAAAGGACATTGCACAGAGACTTATTAGAATTCAACAAAC
GTGTTGGGTCGGGCAAGTCCCCGAAGCTCGGCCAAAAGATTCGCCATGGAACCGTCTGG

Market Trends in Economics





PageRank in Graphs



Equilibrium State

1: 4/13

2: 2/13

3: 2/13

4: 1/13

5: 1/13

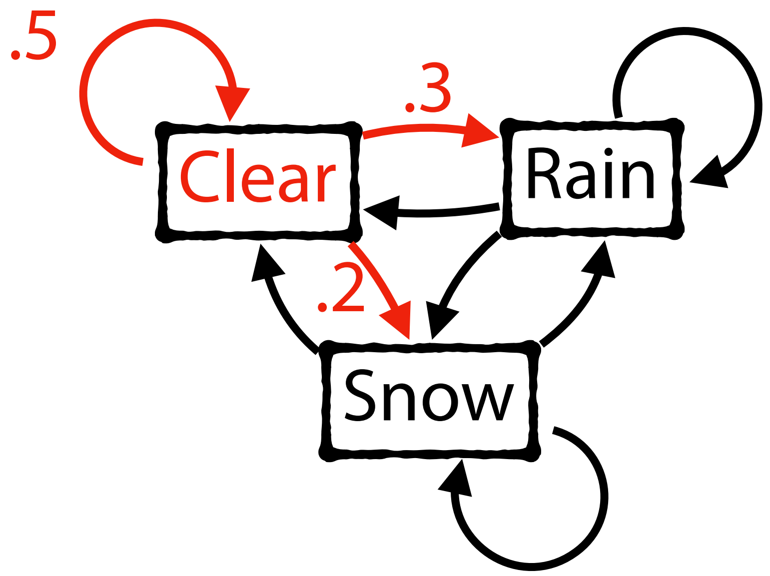
6: 1/13

7: 1/13

8: 1/13

Markov Chain

A **finite Markov Chain** has a set of states S and a finite matrix M

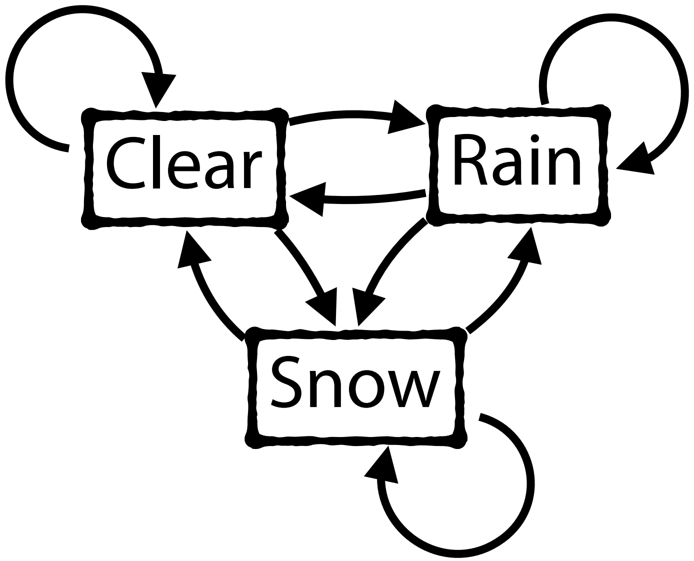


$$S = \{Clear,\ Rain,\ Snow\}$$

$$M = \begin{pmatrix} .5 & .3 & .2 \\ .5 & .4 & .1 \\ .2 & .1 & .7 \end{pmatrix}$$

Markov Chain

Given a Markov Chain and an initial state, all subsequent states can be represented either as **a series of random states** or a transition probability.



$$M = \begin{pmatrix} .5 & .3 & .2 \\ .5 & .4 & .1 \\ .2 & .1 & .7 \end{pmatrix}$$

$X_0 = \text{Clear}$

$X_1 = \text{Clear}$

$X_2 = \text{Snow}$

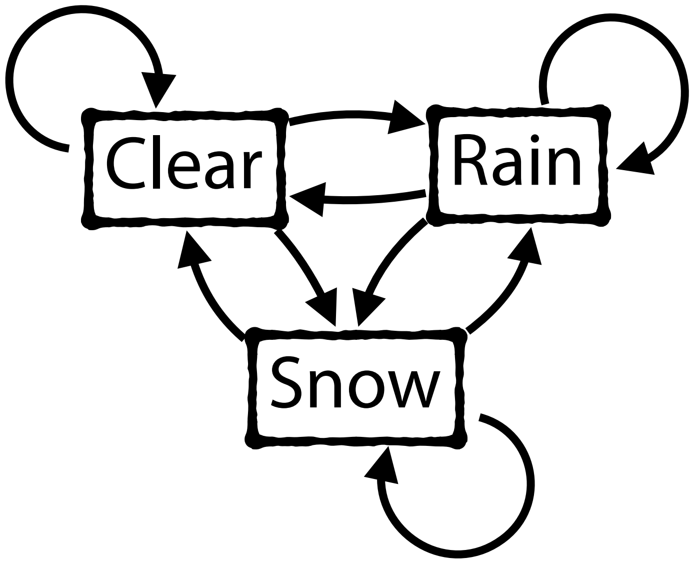
$X_3 = \text{Snow}$

$X_4 = \text{Snow}$

$X_5 = \text{Rain}$

Markov Chain

Given a Markov Chain and an initial state, all subsequent states can be represented either as a series of random states or a **transition probability.**



$$M = \begin{pmatrix} .5 & .3 & .2 \\ .5 & .4 & .1 \\ .2 & .1 & .7 \end{pmatrix}$$

$$M_0 = (.4 \quad .3 \quad .3)$$

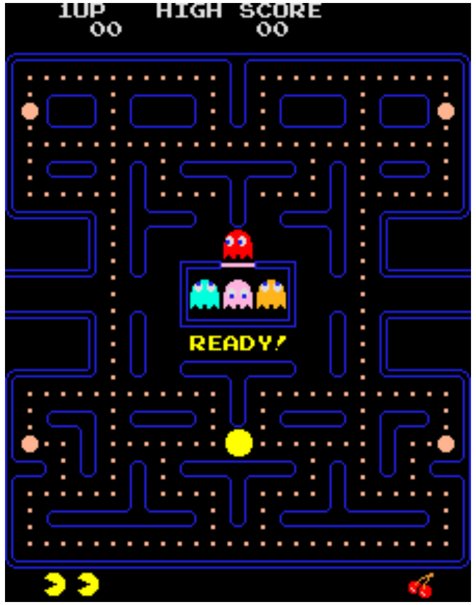
$$M_1 = (.41 \quad .27 \quad .32)$$

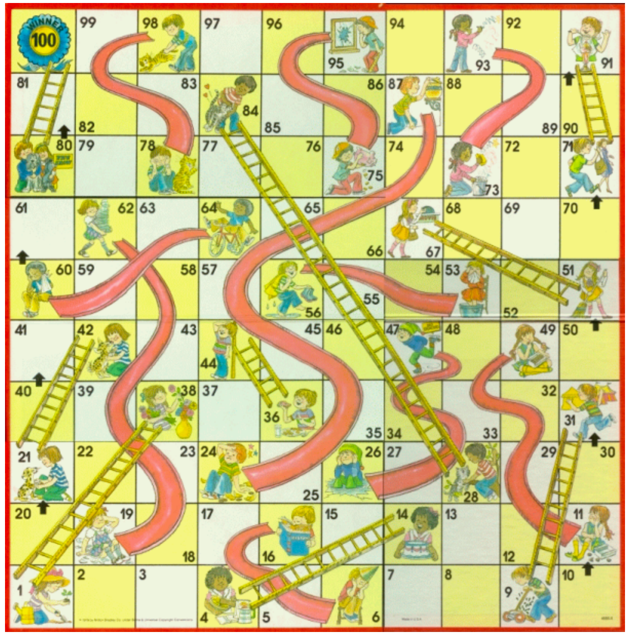
$$M_2 = (.404 \quad .263 \quad .333)$$

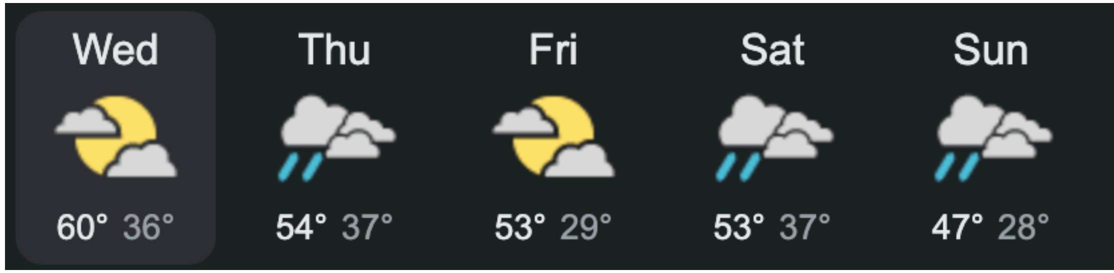
$$M_3 = (.401 \quad .259 \quad .340)$$

Markov Assumption

The probability of the next state depends only on our current state



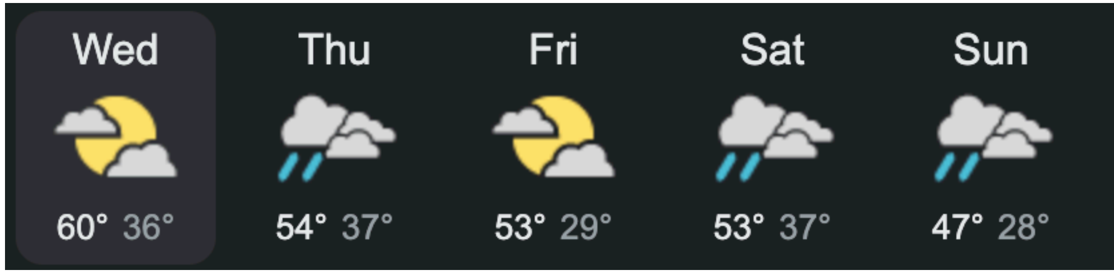




Markov Assumption

Probability of state x_k depends only on previous state x_{k-1}

Ex: Let $x = \{C, R, C, R, R\}$



$$P(x) = P(x_k, x_{k-1}, \ldots x_1)$$

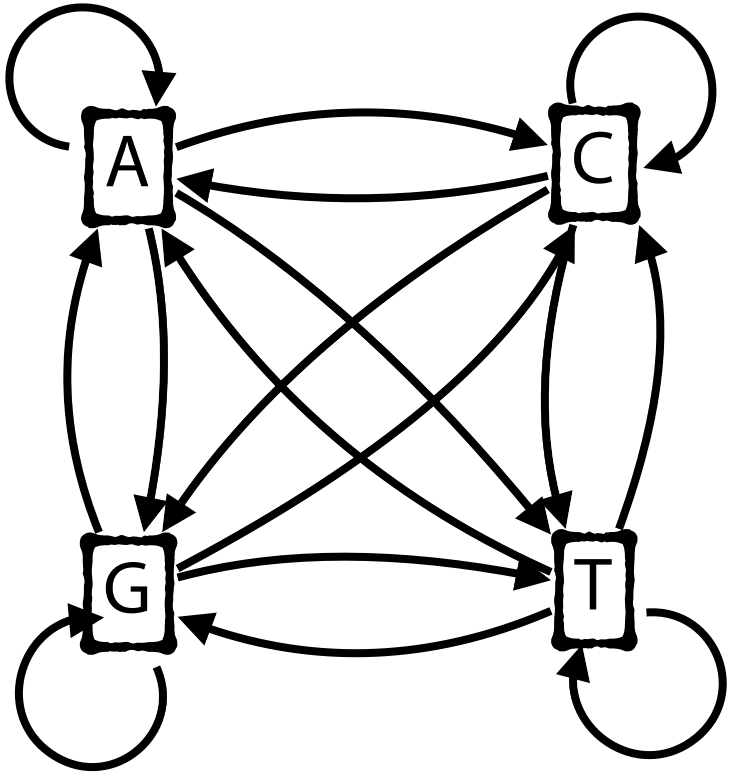
$$= P(x_k \,|\, x_{k-1}, \ldots x_1) P(x_{k-1}, \ldots x_1)$$

$$= P(x_k \,|\, x_{k-1}, \ldots x_1) P(x_{k-1} \,|\, x_{k-2}, \ldots x_1) \ldots P(x_2 \,|\, x_1) P(x_1)$$

$$P(x) \approx$$

Markov Chain in Sequencing

Given a set of sequences, we can construct a model of transitions



P(A | A) = # times AA occurs / # times AX occurs

P(C | A) = # times AC occurs / # times AX occurs

P(G | A) = # times AG occurs / # times AX occurs

P(T | A) = # times AT occurs / # times AX occurs

P(A | C) = # times CA occurs / # times CX occurs

(etc)

where X is any base

Example by Ben Langmead

Markov Chain in Sequencing

Given a set of sequences, we can construct a model of transitions

```
>>> ins_conds, _ = markov_chain_from_dinucs(samp)
>>> print(ins_conds)
[[ 0.19152248,  0.27252589,  0.39998803,  0.1359636 ],
 [ 0.18921984,  0.35832388,  0.25467081,  0.19778547],
 [ 0.17322219,  0.33142737,  0.35571338,  0.13963706],
 [ 0.09509721,  0.33836493,  0.37567927,  0.19085859]]
```

Row labels (X_{i-1}): **A**, **C**, **G**, **T**

Column labels (X_i): **A**, **C**, **G**, **T**

$P(T \mid G)$

Example by Ben Langmead

Markov Chain in Sequencing

```
>>> ins_conds, _ = markov_chain_from_dinucs(samp)
>>> print(ins_conds)
[[ 0.19152248,  0.27252589,  0.39998803,  0.1359636 ],
 [ 0.18921984,  0.35832388,  0.25467081,  0.19778547],
 [ 0.17322219,  0.33142737,  0.35571338,  0.13963706],
 [ 0.09509721,  0.33836493,  0.37567927,  0.19085859]]
```

Row labels (x_{i-1}): A, C, G, T

Column labels (x_i): A, C, G, T

$x = \text{GATC}$

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

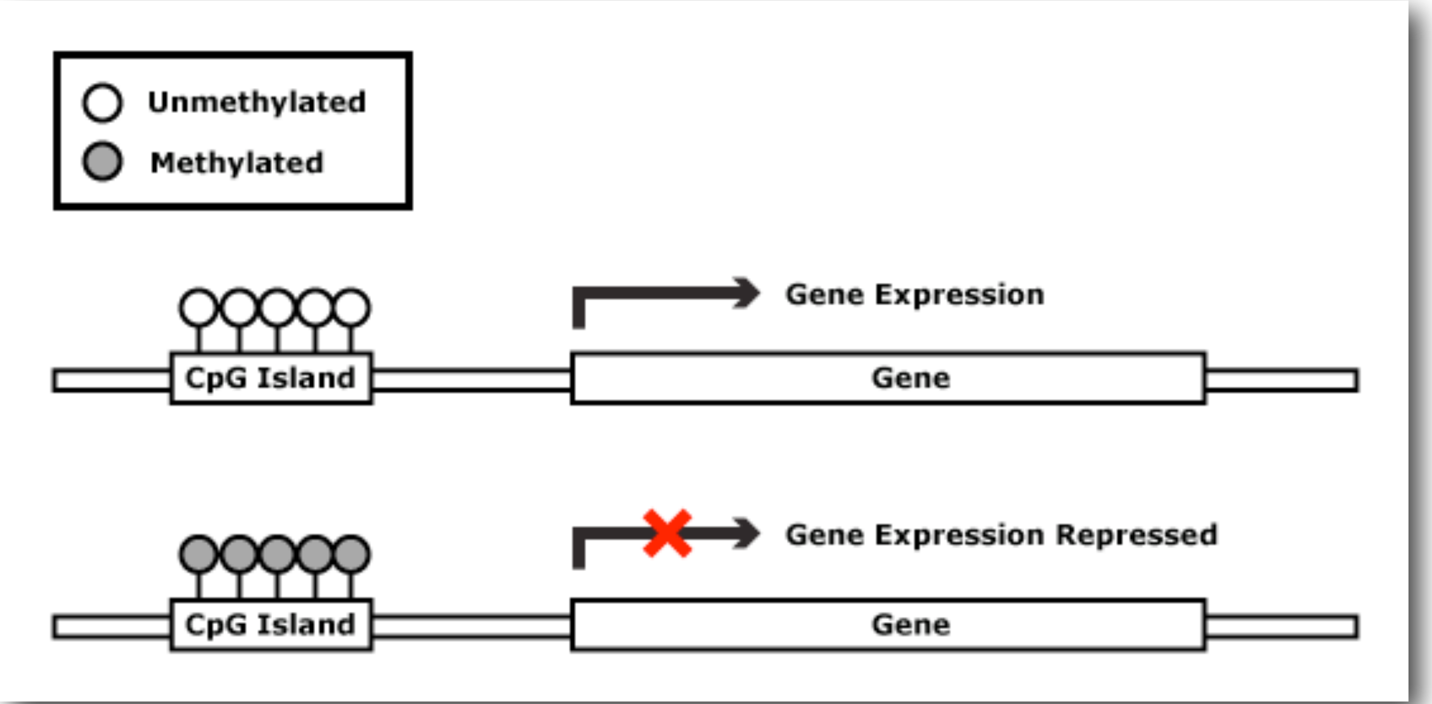
$$P(x) = P(C \mid T)\, P(T \mid A)\, P(A \mid G)\, P(G) = 0.33836493 * 0.1359636 * 0.17322219 * 0.25 = 0.001992$$

Example by Ben Langmead

Markov Chain in Sequencing

We can use this same approach to predict a *label* in our sequences as well

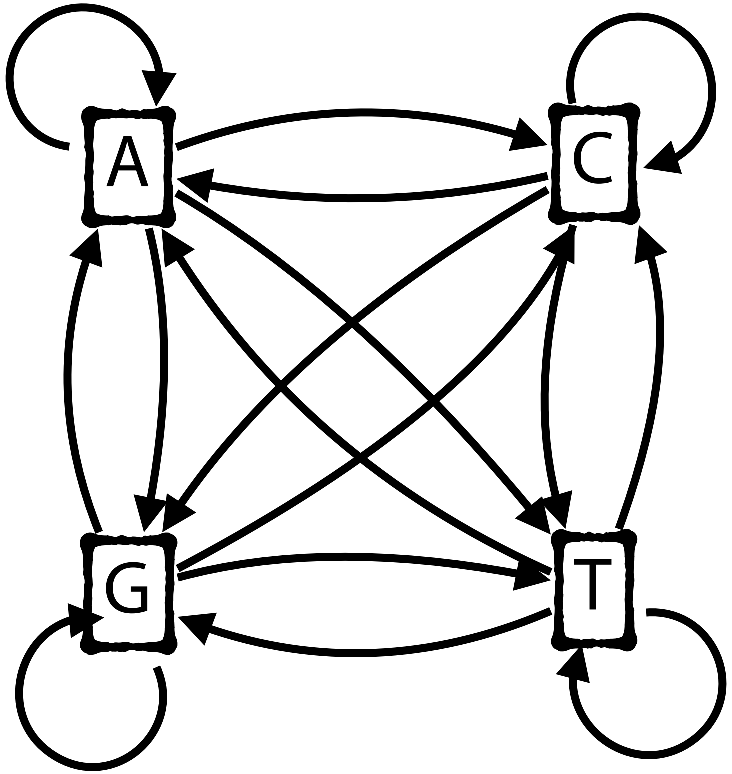
CpG island: part of the genome where CG occurs particularly frequently



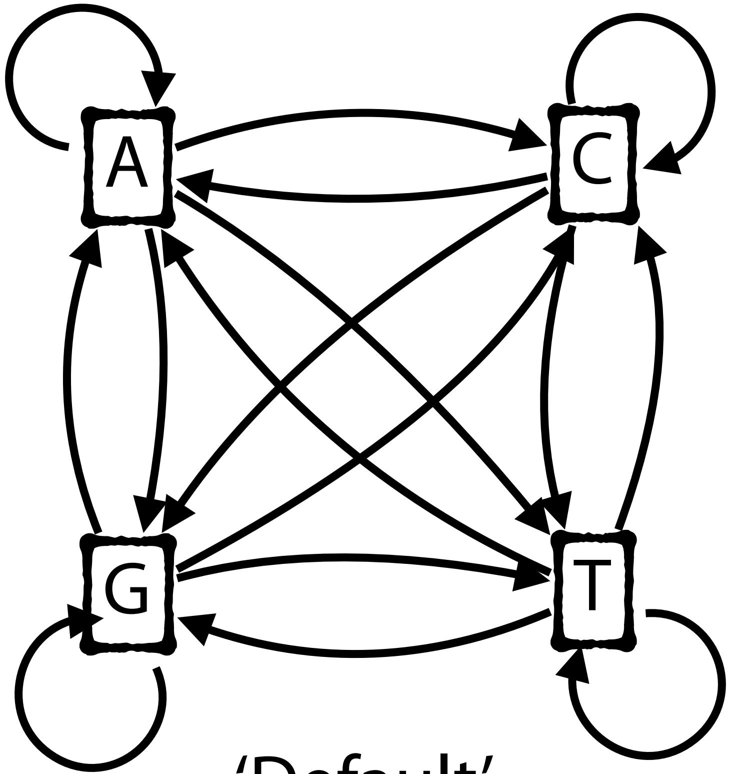
Example by Ben Langmead

Markov Chain in Sequencing

To predict a *label* of a sequencing region, make a Markov chain for both!



CpG Island



'Default'

Example by Ben Langmead

```
>>> cpg_conds, _ = markov_chain_from_dinucs(samp_cpg)
>>> print(cpg_conds)
[[ 0.19152248,  0.27252589,  0.39998803,  0.1359636 ],
 [ 0.18921984,  0.35832388,  0.25467081,  0.19778547],
 [ 0.17322219,  0.33142737,  0.35571338,  0.13963706],
 [ 0.09509721,  0.33836493,  0.37567927,  0.19085859]]
>>> default_conds, _ = markov_chain_from_dinucs(samp_def)
>>> print(default_conds)
[[ 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 ]]
>>> print(np.log2(cpg_conds) - np.log2(def_conds))
[[-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]]
```

CpG: A C G T

Default: A C G T

Log ratio: A C G T

A C G T

Markov Chain in Sequencing

```
>>> print(np.log2(cpg_conds) - np.log2(def_conds))
[[-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]]
```

Rows: x_{i-1} = A, C, G, T; columns: x_i = A, C, G, T

$x = \text{GATC}$

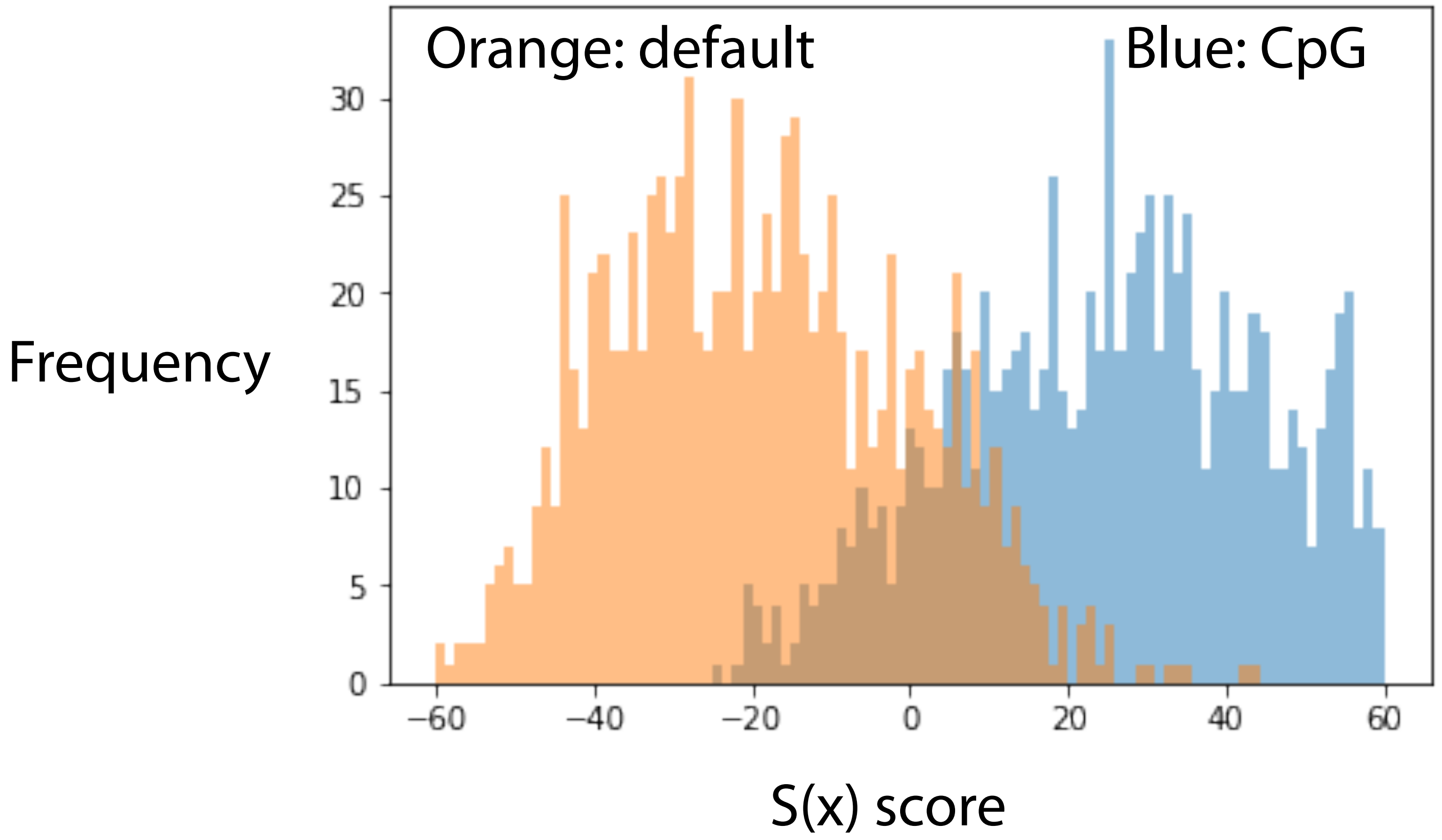
$$P(x) = P(x_4 \mid x_3)\, P(x_3 \mid x_2)\, P(x_2 \mid x_1)\, P(x_1)$$

$$P(x) = P(C \mid T)\, P(T \mid A)\, P(A \mid G)\, P(G) = 0.73036913 + -0.85527103 + -0.79486196 = -0.919763$$

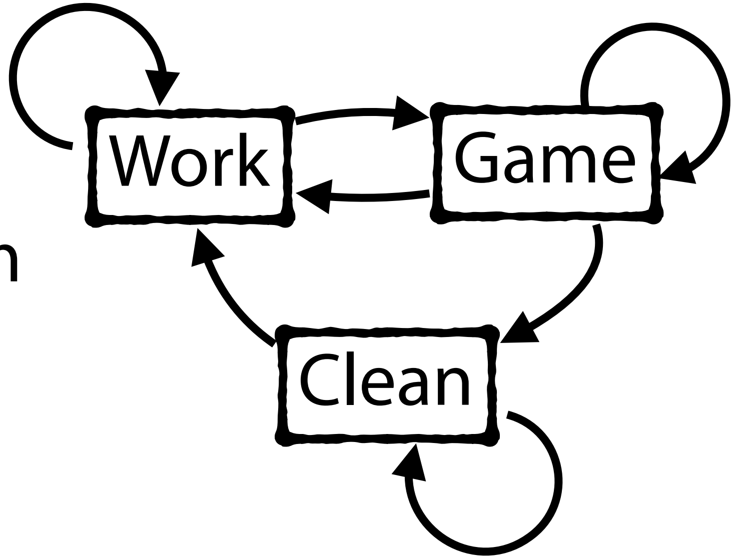
Example by Ben Langmead

Markov Chain in Sequencing

Drew 1,000 100-mers from inside CpG islands and another 1,000 from outside, and calculated S(x) for all



Markov Chain Matrix

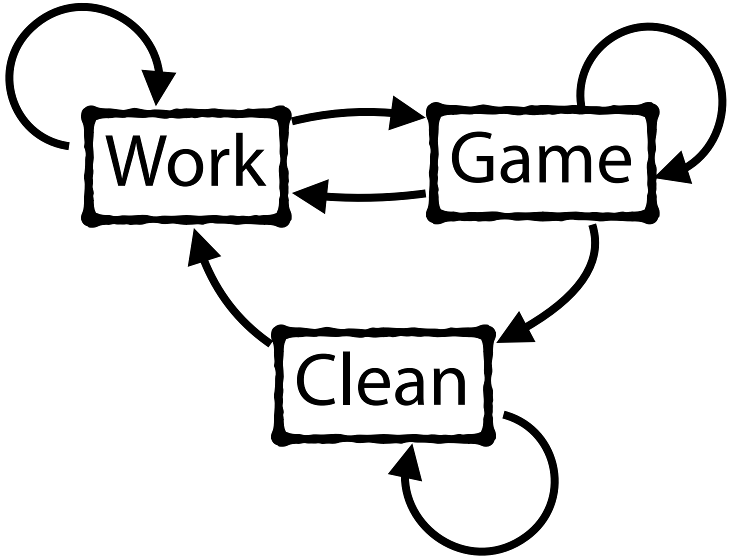


If I'm working at time 0, what is probability that I'm working at time t?

Claim: $Pr(X_t = v \mid X_0 = u) = M^t[u, v]$

$$M = \begin{pmatrix} .4 & .6 & 0 \\ .1 & .6 & .3 \\ .5 & 0 & .5 \end{pmatrix}$$

Markov Chain Matrix



Claim: $Pr(X_t = v \mid X_0 = u) = M^t[u, v]$

Base Case:

T=1:

$$M = \begin{pmatrix} .4 & .6 & 0 \\ .1 & .6 & .3 \\ .5 & 0 & .5 \end{pmatrix}$$

T=2:

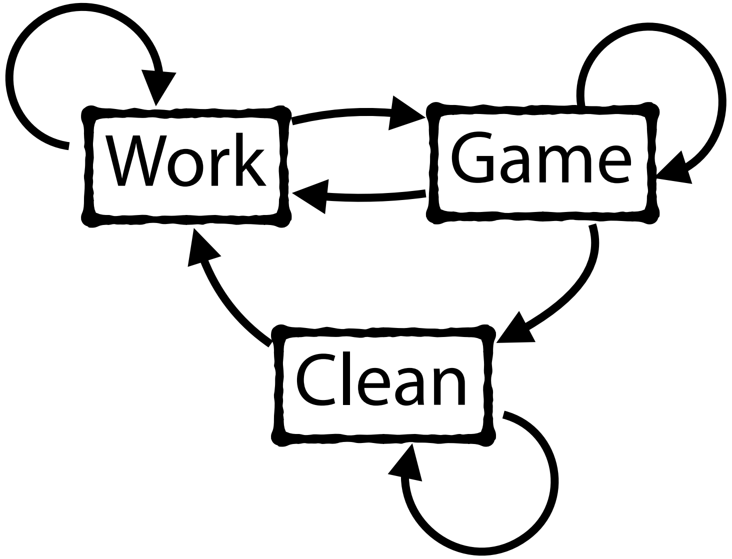
Markov Chain Matrix

Claim: $Pr(X_t = v \mid X_0 = u) = M^t[u, v]$

Induction:

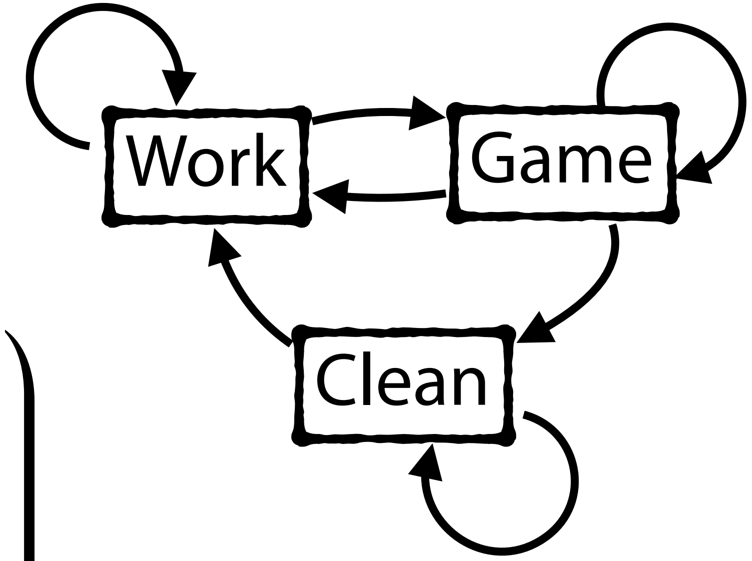
Assume $Pr(X_{t-1} = v \mid X_0 = u) = M^{t-1}[u, v]$.

Show holds for $Pr(X_t = w \mid X_0 = u) = M^t[u, w]$



$$M = \begin{pmatrix} .4 & .6 & 0 \\ .1 & .6 & .3 \\ .5 & 0 & .5 \end{pmatrix}$$

Markov Chain Matrix



What happens as $t \to \infty$?

$$M = \begin{pmatrix} .4 & .6 & 0 \\ .1 & .6 & .3 \\ .5 & 0 & .5 \end{pmatrix} \qquad M^3 = \begin{pmatrix} .238 & .492 & .270 \\ .307 & .402 & .291 \\ .335 & .450 & .215 \end{pmatrix}$$

$$M^{10} = \begin{pmatrix} .2940 & .4413 & .2648 \\ .2942 & .4411 & .2648 \\ .2942 & .4413 & .2648 \end{pmatrix}$$

$$M^{60} = \begin{pmatrix} .2941 & .4412 & .2647 \\ .2941 & .4412 & .2647 \\ .2941 & .4412 & .2647 \end{pmatrix}$$

Markov Chain Stationary Distribution

A probability vector π is called a **stationary distribution** for a Markov Chain if it satisfies the stationary equation: $\pi = \pi M$

$$M = \begin{pmatrix} .4 & .6 & 0 \\ .1 & .6 & .3 \\ .5 & 0 & .5 \end{pmatrix}$$

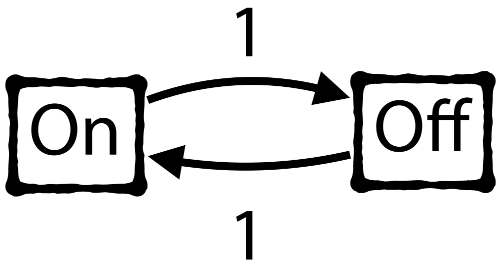
$$\pi[W] = .4\pi[W] + .1\pi[G] + .5\pi[C]$$

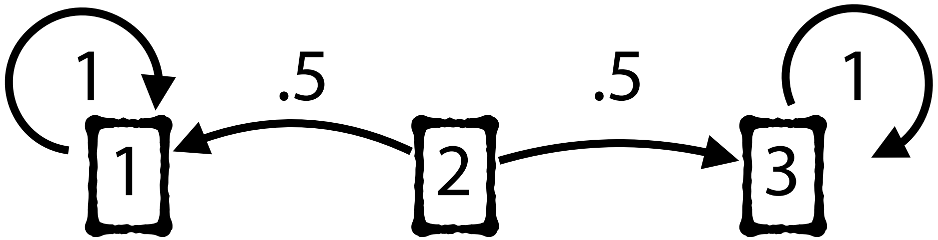
$$\pi[S] = .6\pi[W] + .6\pi[G] + 0\pi[C]$$

$$\pi[E] = 0\pi[W] + .3\pi[G] + .5\pi[C]$$

Markov Chain Stationary Distribution

Stationary distributions can be calculated using the system of equation (and that all probabilities sum to 1). **But not every Markov Chain has a steady state (and some have infinitely many)!**





Markov Chain Monte Carlo

There are ways to prove whether a Markov Chain has a stationary distribution, but several algorithms exist that approximate!

Gibbs Sampling:

Randomly assign values to a probability vector $\pi = (\theta_1, \theta_2, \ldots, \theta_d)$.

Repeatedly:

Pick a random $1 \leq i \leq d$

Randomly update value $\theta_i \,|\, \theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_d$

Markov Chain Monte Carlo

There are ways to prove whether a Markov Chain has a stationary distribution, but several algorithms exist that approximate!

Gibbs Sampling:

Randomly assign values to a probability vector $\pi = (\theta_1, \theta_2, \ldots, \theta_d)$.

Repeatedly:

Pick a random $1 \leq i \leq d$

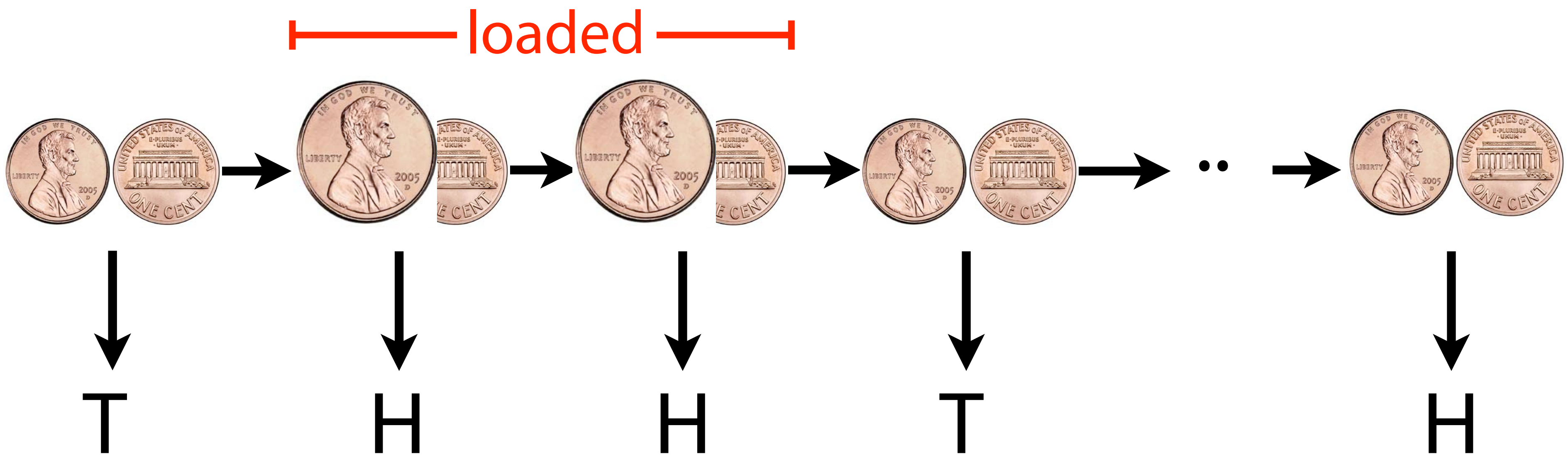
Randomly update value θ_i based on $\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_d$

Hidden Markov Models

In the real world, we often don't know the underlying markov chain!

Instead, we have observations that can be used to predict our current state.

Ex: Repeated coin flips but *sometimes* I cheat and use a fixed coin.



Hidden Markov Model