Markov chains and the number of occurrences of a word in a sequence (4.5–4.9, 11.1,2,4,6)

Prof. Tesler

Math 283
Fall 2018

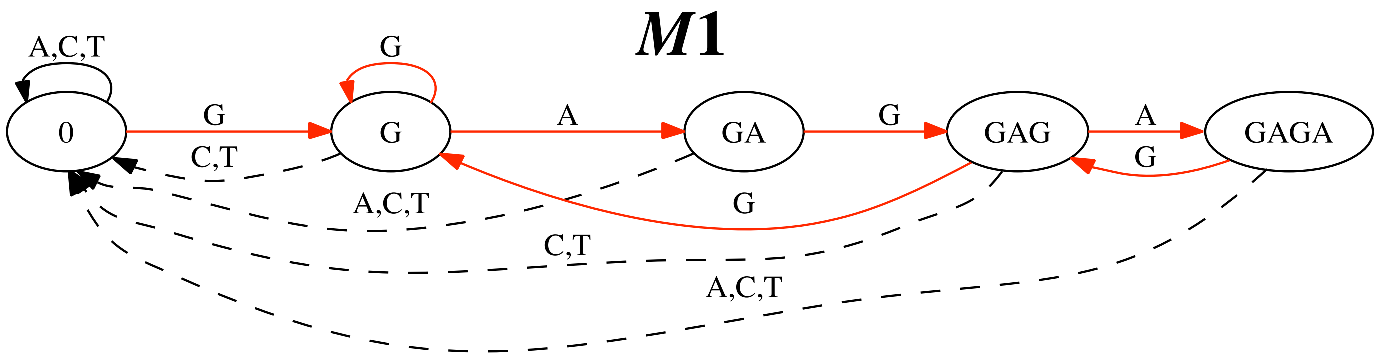
Locating overlapping occurrences of a word

- Consider a (long) single-stranded nucleotide sequence $\tau = \tau_1 \ldots \tau_N$ and a (short) word $w = w_1 \ldots w_k$, e.g., $w =$ `GAGA`.

```
for i = 1 to N-3 {
    if (τ_i τ_{i+1} τ_{i+2} τ_{i+3} == GAGA) {
        ...
    }
}
```

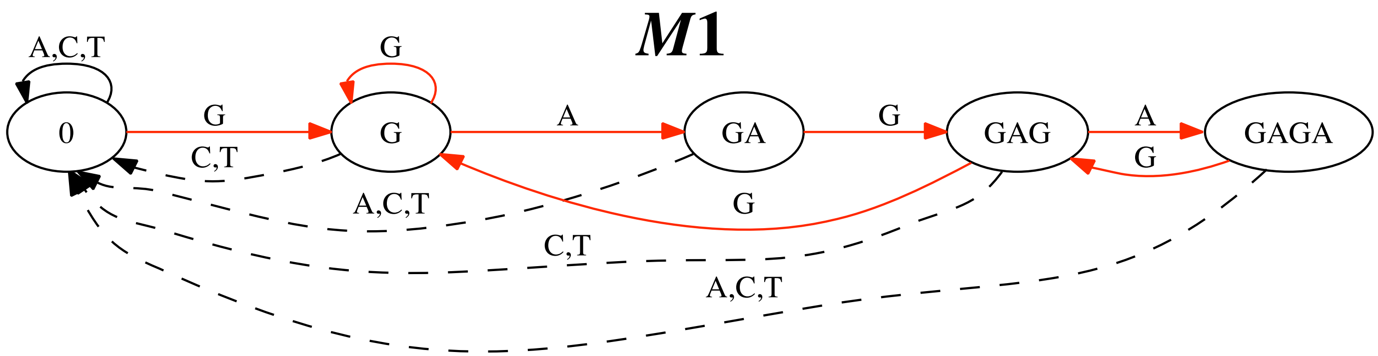
- The above scan takes up to $\approx 4N$ comparisons to locate all occurrences of `GAGA` (kN comparisons for w of length k).
- A *finite state automaton* (FSA) is a "machine" that can locate all occurrences while only examining each letter of τ *once*.

Overlapping occurrences of GAGA



- The *states* are the nodes $\emptyset$, G, GA, GAG, GAGA (prefixes of w).
  For $w = w_1 w_2 \cdots w_k$, there are $k + 1$ states (one for each prefix).
- Start in the state $\emptyset$ (shown on figure as 0).
- Scan $\tau = \tau_1 \tau_2 \ldots \tau_N$ one character at a time left to right.
- *Transition edges:* When examining τ_j, move from the current state to the next state according to which edge τ_j is on.
  - For each node $u = w_1 \cdots w_r$ and each letter $x =$ A, C, G, T, determine the longest suffix s (possibly $\emptyset$) of $w_1 \cdots w_r x$ that's among the states.
  - Draw an edge $u \xrightarrow{x} s$
- The number of times we are in the state GAGA is the desired count of number of occurrences.

Overlapping occurrences of GAGA in $\tau =$ CAGAGGTCGAGAGT...

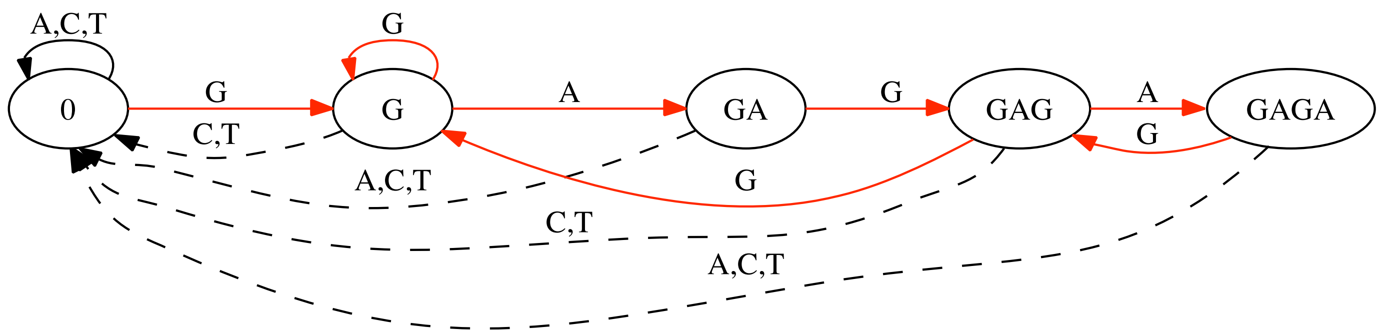


t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
τ_t	C	A	G	A	G	G	T	C	G	A	G	A	G	T	...	

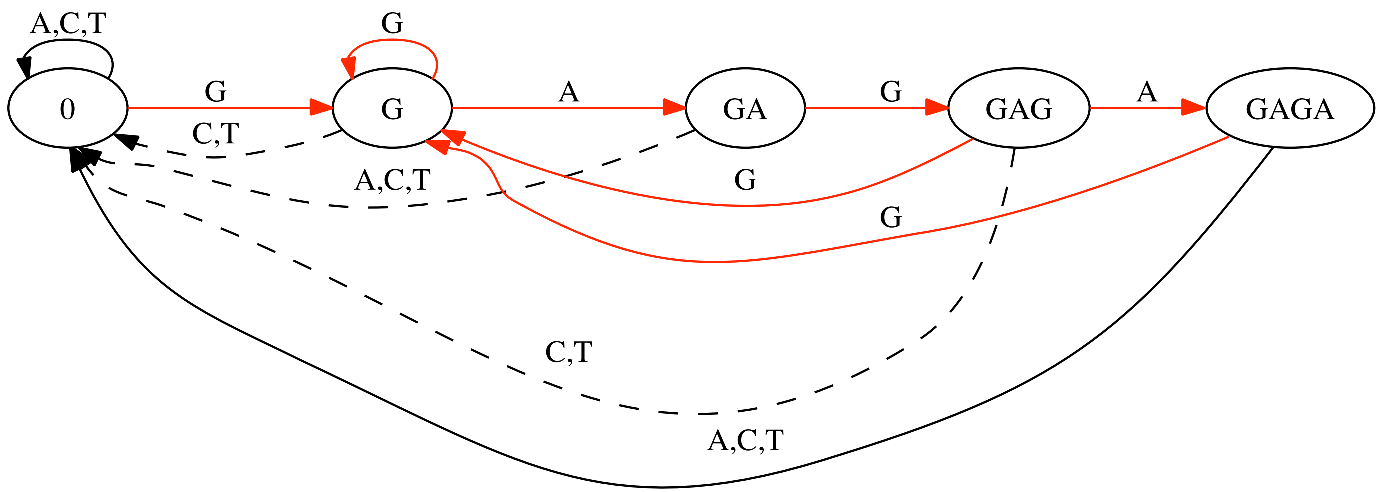
Time t	State at t	τ_t
1	0	C
2	0	A
3	0	G
4	G	A
5	GA	G
6	GAG	G
7	G	T
8	0	C
9	0	G
10	G	A
11	GA	G
12	GAG	A
13	GAGA	G
14	GAG	T
15	0	$\cdots$

Non-overlapping occurrences of GAGA

$M1 =$

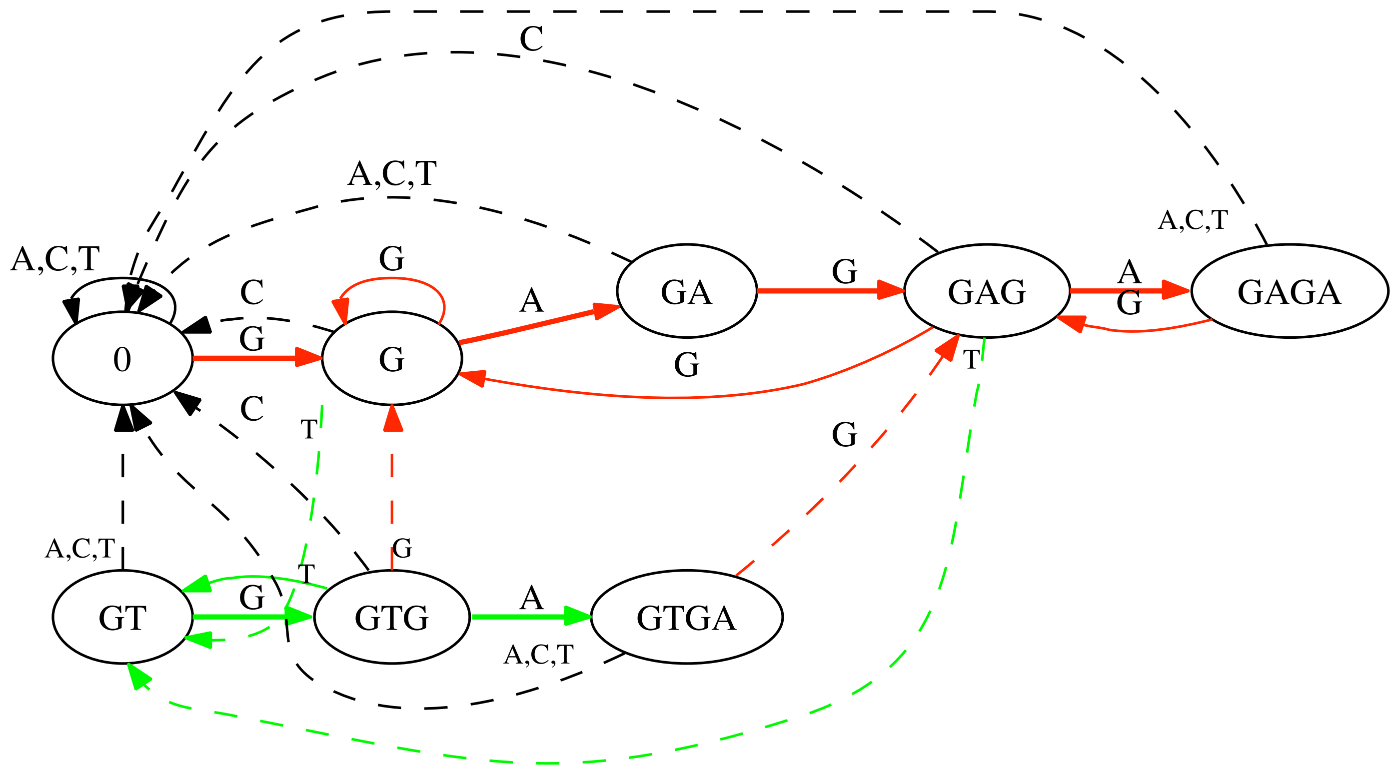


$M2 =$



- For non-overlapping occurrences of w:
  - Replace the outgoing edges from w by copies of the outgoing edges from $\emptyset$.
- On previous slide, the time $13 \to 14$ transition $\texttt{GAGA} \xrightarrow{\texttt{G}} \texttt{GAG}$ changes to $\texttt{GAGA} \xrightarrow{\texttt{G}} \texttt{G}$.

Motif {GAGA, GTGA}, overlaps permitted



- *States:* All prefixes of all words in the motif.
  If a prefix occurs multiple times, only create one node for it.
- *Transition edges:* they may jump from one word of the motif to another.
  - GTGA $\xrightarrow{\text{G}}$ GAG.
- Count the number of times we reach the states for any words in the motif (GAGA or GTGA).

Markov chains

- A Markov chain is similar to a finite state machine, but incorporates probabilities.
- Let S be a set of “states.”
  We will take S to be a discrete finite set, such as $S = \{1, 2, \ldots, s\}$.
- Let $t = 1, 2, \ldots$ denote the “time.”
- Let $X_1, X_2, \ldots$ denote a sequence of random variables, values $\in S$.

The X_t’s form a *(first order) Markov chain* if they obey these rules

1. The probability of being in a certain state at time $t + 1$ only depends on the state at time t, not on any earlier states:
   $$P(X_{t+1} = x_{t+1} | X_1 = x_1, \ldots, X_t = x_t) = P(X_{t+1} = x_{t+1} | X_t = x_t)$$
2. The probability of transitioning from state i at time t to state j at time $t + 1$ only depends on i and j, *but not on the time t:*
   $$P(X_{t+1} = j | X_t = i) = p_{ij} \text{ at } \textit{all} \text{ times } t$$
   for some values p_{ij}, which form an $s \times s$ *transition matrix*.

Transition matrix

The *transition matrix*, $P1$, of the Markov chain $M1$ is

From state To state 1 2 3 4 5

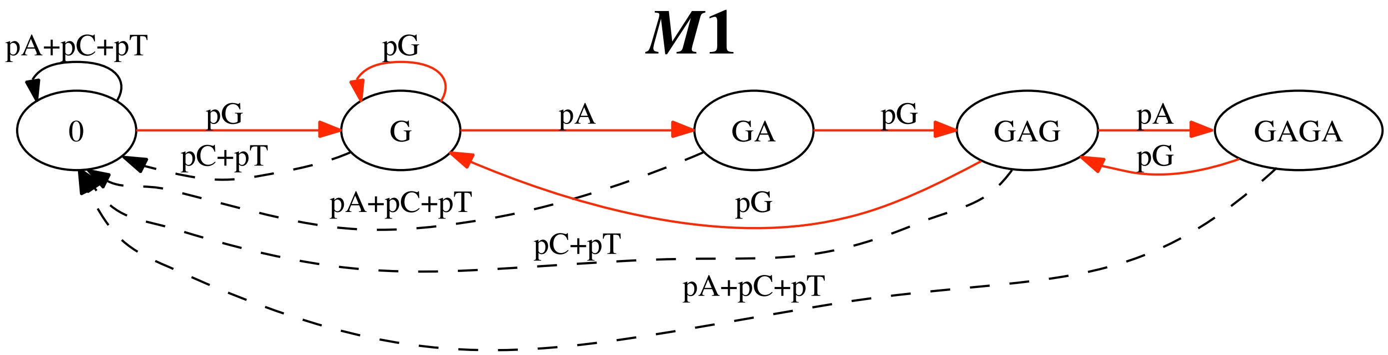
From state	To state 1	2	3	4	5
1: 0	$p_A+p_C+p_T$	p_G	0	0	0
2: G	p_C+p_T	p_G	p_A	0	0
3: GA	$p_A+p_C+p_T$	0	0	p_G	0
4: GAG	p_C+p_T	p_G	0	0	p_A
5: GAGA	$p_A+p_C+p_T$	0	0	p_G	0

$$\begin{bmatrix} p_A+p_C+p_T & p_G & 0 & 0 & 0 \\ p_C+p_T & p_G & p_A & 0 & 0 \\ p_A+p_C+p_T & 0 & 0 & p_G & 0 \\ p_C+p_T & p_G & 0 & 0 & p_A \\ p_A+p_C+p_T & 0 & 0 & p_G & 0 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} \\ P_{41} & P_{42} & P_{43} & P_{44} & P_{45} \\ P_{51} & P_{52} & P_{53} & P_{54} & P_{55} \end{bmatrix}$$

- Notice that the entries in each row sum up to $p_A+p_C+p_G+p_T=1$.
- A matrix with all entries $\geqslant 0$ and all row sums equal to 1 is called a *stochastic matrix*.
- The transition matrix of a Markov chain is always stochastic.
- All row sums $=1$ can be written

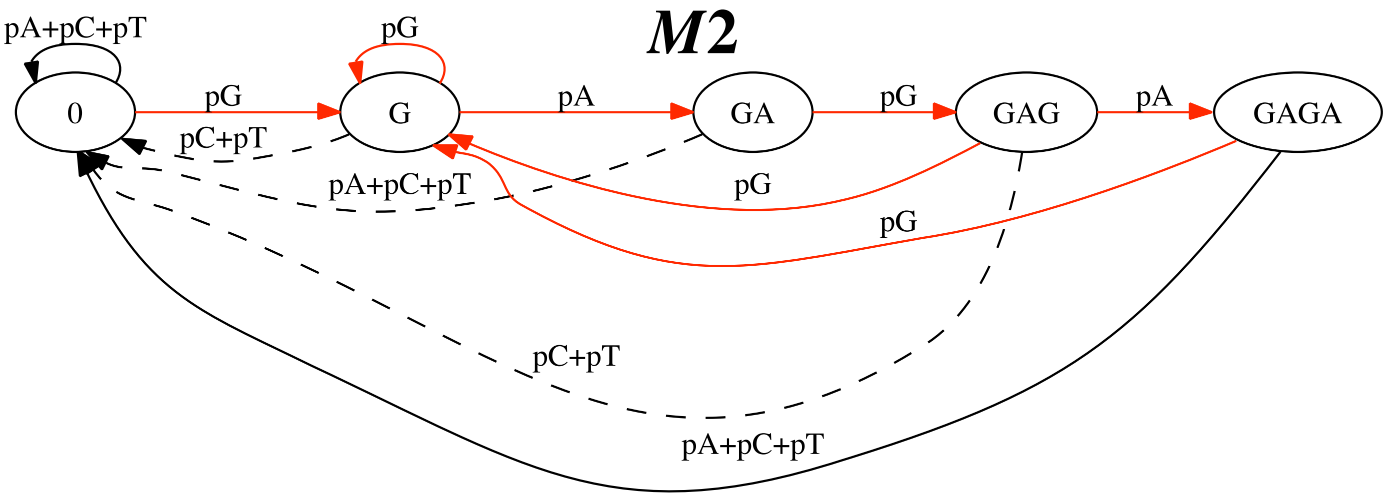
$$P\vec{1}=\vec{1} \qquad \text{where } \vec{1}=\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}$$

so $\vec{1}$ is a right eigenvector of P with eigenvalue 1.



***P*1**

$$\begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 0 & 0 & 1/4 & 0 \end{bmatrix}$$



***P*2**

$$\begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 1/4 & 0 & 0 & 0 \end{bmatrix}$$

- Edge labels are replaced by probabilities, e.g., $p_C + p_T$.
- The matrices are shown for the case that all nucleotides have equal probabilities $1/4$.
- $P2$ (no overlaps) is obtained from $P1$ (overlaps allowed) by replacing the last row with a copy of the first row.

Other applications of automata

- Automata / state machines are also used in other applications in Math and Computer Science. The transition weights may be defined differently, and the matrices usually aren't stochastic.
- **Combinatorics:** Count walks through the automaton (instead of getting their probabilities) by setting transition weights $u \xrightarrow{x} s$ to 1.
- **Computer Science (formal languages, classifiers, . . .):** Does the string τ contain `GAGA`? Output 1 if it does, 0 otherwise.
  - Modify $M1$: remove the outgoing edges on `GAGA`.
  - On reaching state `GAGA`, terminate with output 1.
  - If the end of τ is reached, terminate with output 0.
  - This is called a *deterministic finite acceptor* (DFA).
- **Markov chains:** Instead of considering a specific string τ, we'll compute probabilities, expected values, . . . over the sample space of all strings of length n.

Other Markov chain examples

- A Markov chain is kth order if the probability of $X_t = i$ depends on the values of $X_{t-1}, \ldots, X_{t-k}$. It can be converted to a first order Markov chain by making new states that record more history.

- *Positional independence*: Instead of a null hypothesis that a DNA sequence is generated by repeated rolls of a biased four-sided die, we could use a Markov chain. The simplest is a *one-step transition matrix*

$$P = \begin{bmatrix} p_{\mathtt{AA}} & p_{\mathtt{AC}} & p_{\mathtt{AG}} & p_{\mathtt{AT}} \\ p_{\mathtt{CA}} & p_{\mathtt{CC}} & p_{\mathtt{CG}} & p_{\mathtt{CT}} \\ p_{\mathtt{GA}} & p_{\mathtt{GC}} & p_{\mathtt{GG}} & p_{\mathtt{GT}} \\ p_{\mathtt{TA}} & p_{\mathtt{TC}} & p_{\mathtt{TG}} & p_{\mathtt{TT}} \end{bmatrix}$$

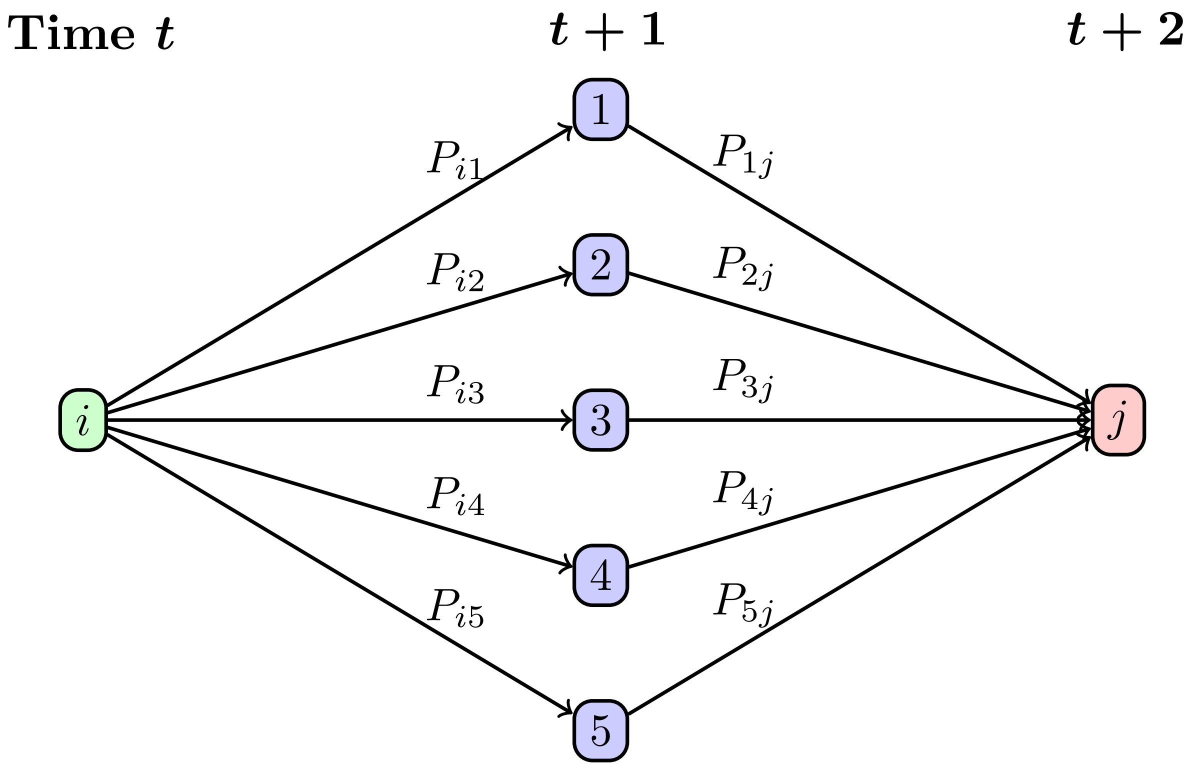
  P could be the same at all positions. In a coding region, it could be different for the first, second, and third positions of codons.

- *Nucleotide evolution:* There are models of random point mutations over the course of evolution concerning Markov chains with the form P (same as above) in which X_t is the state $\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}$ of the nucleotide at a given position in a sequence at time (generation) t.

Questions about Markov chains

1. What is the probability of being in a particular state after n steps?
2. What is the probability of being in a particular state as $n \to \infty$?
3. What is the "reverse" Markov chain?
4. If you are in state i, what is the expected number of time steps until the next time you are in state j? What is the variance of this? What is the complete probability distribution?
5. Starting in state i, what is the expected number of visits to state j before reaching state k?

Transition probabilities after two steps



To compute the probability for going from state i at time t to state j at time $t+2$, consider all the states it could go through at time $t+1$:

$$\begin{aligned} P(X_{t+2}=j|X_t=i) &= \sum\nolimits_r P(X_{t+1}=r|X_t=i)P(X_{t+2}=j|X_{t+1}=r, X_t=i) \\ &= \sum\nolimits_r P(X_{t+1}=r|X_t=i)P(X_{t+2}=j|X_{t+1}=r) \\ &= \sum\nolimits_r P_{ir}P_{rj} = (P^2)_{ij} \end{aligned}$$

Transition probabilities after n steps

For $n \geqslant 0$, the transition matrix from time t to time $t+n$ is P^n:

$$P(X_{t+n} = j | X_t = i) = \sum_{r_1,\ldots,r_{n-1}} P(X_{t+1} = r_1 | X_t = i) P(X_{t+2} = r_2 | X_{t+1} = r_1) \cdots$$

$$= \sum_{r_1,\ldots,r_{n-1}} P_{i\,r_1} P_{r_1\,r_2} \cdots P_{r_{n-1}\,j} \;=\; (P^n)_{ij}$$

(sum over possible states $r_1, \ldots, r_{n-1}$ at times $t+1, \ldots, t+(n-1)$)

State probability vector

- $\alpha_i(t) = P(X_t = i)$ is the probability of being in state i at time t.
- Column vector $\vec{\alpha}(t) = \begin{pmatrix} \alpha_1(t) \\ \vdots \\ \alpha_s(t) \end{pmatrix}$
  or transpose it to get a row vector $\vec{\alpha}(t)' = (\alpha_1(t), \ldots, \alpha_s(t))$
- The probabilities at time $t+n$ are

$$\begin{aligned} \alpha_j(t+n) &= P(X_{t+n} = j | \vec{\alpha}(t)) = \sum_i P(X_{t+n} = j | X_t = i) P(X_t = i) \\ &= \sum_i \alpha_i(t) (P^n)_{ij} = (\vec{\alpha}(t)' P^n)_j \end{aligned}$$

  so $\vec{\alpha}(t+n)' = \vec{\alpha}(t)' P^n$ (row vector times matrix)
  or equivalently, $(P')^n \vec{\alpha}(t) = \vec{\alpha}(t+n)$ (matrix times column vector).

State vector after n steps for GAGA; $P = P1$

$$P = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \end{bmatrix} \qquad P^2 = \begin{bmatrix} \frac{11}{16} & \frac{1}{4} & \frac{1}{16} & 0 & 0 \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \end{bmatrix} \qquad (P')^2 = \begin{bmatrix} \frac{11}{16} & \frac{11}{16} & \frac{11}{16} & \frac{11}{16} & \frac{11}{16} \\ \frac{1}{4} & \frac{3}{16} & \frac{1}{4} & \frac{3}{16} & \frac{1}{4} \\ \frac{1}{16} & \frac{1}{16} & 0 & \frac{1}{16} & 0 \\ 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 \\ 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} \end{bmatrix}$$

- At $t = 10$, suppose $\frac{1}{3}$ chance of being in the 1$^{\text{st}}$ state; $\frac{2}{3}$ chance of being in the 2$^{\text{nd}}$ state; and no chance of other states:

$$\vec{\alpha}(10)' = \left(\tfrac{1}{3}, \tfrac{2}{3}, 0, 0, 0\right).$$

- Time $t = 12$ is $n = 12 - 10 = 2$ steps later:

$$\vec{\alpha}(12)' = \left(\tfrac{1}{3}, \tfrac{2}{3}, 0, 0, 0\right)P^2 = \left(\tfrac{11}{16}, \tfrac{5}{24}, \tfrac{1}{16}, \tfrac{1}{24}, 0\right)$$

- Alternately:

$$\vec{\alpha}(10) = \begin{pmatrix} 1/3 \\ 2/3 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \vec{\alpha}(2) = (P')^2 \vec{\alpha}(10) = \begin{pmatrix} 11/16 \\ 5/24 \\ 1/16 \\ 1/24 \\ 0 \end{pmatrix}$$

Transition probabilities after n steps for GAGA; $P = P1$

$$P^0 = I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad P^1 = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \end{bmatrix} \qquad P^2 = \begin{bmatrix} \frac{11}{16} & \frac{1}{4} & \frac{1}{16} & 0 & 0 \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \\ \frac{11}{16} & \frac{3}{16} & \frac{1}{16} & \frac{1}{16} & 0 \\ \frac{11}{16} & \frac{1}{4} & 0 & 0 & \frac{1}{16} \end{bmatrix}$$

$$P^3 = \begin{bmatrix} \frac{11}{16} & \frac{15}{64} & \frac{1}{16} & \frac{1}{64} & 0 \\ \frac{11}{16} & \frac{15}{64} & \frac{3}{64} & \frac{1}{64} & \frac{1}{64} \\ \frac{11}{16} & \frac{15}{64} & \frac{1}{16} & \frac{1}{64} & 0 \\ \frac{11}{16} & \frac{15}{64} & \frac{3}{64} & \frac{1}{64} & \frac{1}{64} \\ \frac{11}{16} & \frac{15}{64} & \frac{1}{16} & \frac{1}{64} & 0 \end{bmatrix} \qquad P^4 = \begin{bmatrix} \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \\ \frac{11}{16} & \frac{15}{64} & \frac{15}{256} & \frac{1}{64} & \frac{1}{256} \end{bmatrix} \qquad P^n = P^4 \text{ for } n \geqslant 5$$

- Regardless of the starting state, the probabilities of being in states $1, \cdots, 5$ at time t (when t is large enough) are $\frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256}$.
- Usually P^n just approaches a limit asymptotically as n increases, rather than reaching it. We'll see other examples later (like $P2$).

Matrix powers in Matlab and R

Matlab

```
>> P1 = [
 [ 3/4, 1/4,   0,   0,   0 ];  %
 [ 2/4, 1/4, 1/4,   0,   0 ];  % G
 [ 3/4,   0,   0, 1/4,   0 ];  % GA
 [ 2/4, 1/4,   0,   0, 1/4 ];  % GAG
 [ 3/4,   0,   0, 1/4,   0 ];  % GAGA
]

P1 =
  0.7500  0.2500       0       0       0
  0.5000  0.2500  0.2500       0       0
  0.7500       0       0  0.2500       0
  0.5000  0.2500       0       0  0.2500
  0.7500       0       0  0.2500       0

>> P1 * P1      % or    P1^2
ans =
  0.6875  0.2500  0.0625       0       0
  0.6875  0.1875  0.0625  0.0625       0
  0.6875  0.2500       0       0  0.0625
  0.6875  0.1875  0.0625  0.0625       0
  0.6875  0.2500       0       0  0.0625
```

R

```
> P1 = rbind(
+   c(3/4,1/4,  0,  0,  0), #
+   c(2/4,1/4,1/4,  0,  0), # G
+   c(3/4,  0,  0,1/4,  0), # GA
+   c(2/4,1/4,  0,  0,1/4), # GAG
+   c(3/4,  0,  0,1/4,  0)  # GAGA
+ )

> P1
     [,1] [,2] [,3] [,4] [,5]
[1,] 0.75 0.25 0.00 0.00 0.00
[2,] 0.50 0.25 0.25 0.00 0.00
[3,] 0.75 0.00 0.00 0.25 0.00
[4,] 0.50 0.25 0.00 0.00 0.25
[5,] 0.75 0.00 0.00 0.25 0.00

> P1 %*% P1
       [,1]   [,2]   [,3]   [,4]   [,5]
[1,] 0.6875 0.2500 0.0625 0.0000 0.0000
[2,] 0.6875 0.1875 0.0625 0.0625 0.0000
[3,] 0.6875 0.2500 0.0000 0.0000 0.0625
[4,] 0.6875 0.1875 0.0625 0.0625 0.0000
[5,] 0.6875 0.2500 0.0000 0.0000 0.0625
```

Note: R doesn't have a built-in matrix power function.
The $>$ and $+$ symbols above are prompts, not something you enter.

Stationary distribution, a.k.a. steady state distribution

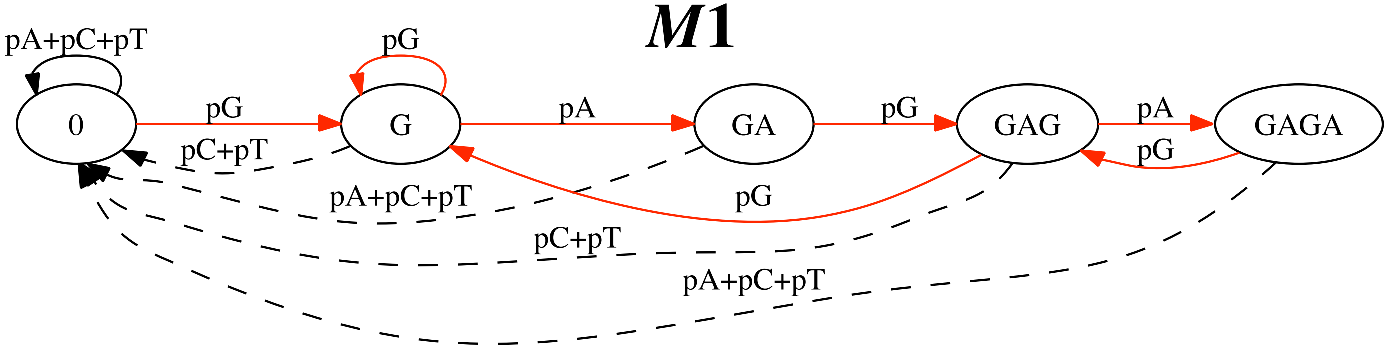
- If P is irreducible and aperiodic (these will be defined soon) then P^n approaches a limit with this format as $n \to \infty$:

$$\lim_{n\to\infty} P^n = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \end{bmatrix}$$

- In other words, no matter what the starting state, the probability of being in state j after n steps approaches φ_j.
- The row vector $\vec{\varphi}\,' = (\varphi_1, \ldots, \varphi_s)$ is called the *stationary distribution* of the Markov chain.
- It is "stationary" because these probabilities stay the same from one time to the next; in matrix notation, $\vec{\varphi}\,' P = \vec{\varphi}\,'$, or $P' \vec{\varphi} = \vec{\varphi}$.
- So $\vec{\varphi}\,'$ is a left eigenvector of P with eigenvalue 1.
- Since it represents probabilities of being in each state, the components of $\vec{\varphi}$ add up to 1.

Stationary distribution — computing it for example $M1$

$M1$



$P1$

$$\begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 0 & 0 & 1/4 & 0 \end{bmatrix}$$

- Solve $\vec{\varphi}\,'P = \vec{\varphi}\,'$, or $(\varphi_1, \ldots, \varphi_5)P = (\varphi_1, \ldots, \varphi_5)$:

$$\begin{aligned}
\varphi_1 &= \tfrac{3}{4}\varphi_1 + \tfrac{1}{2}\varphi_2 + \tfrac{3}{4}\varphi_3 + \tfrac{1}{2}\varphi_4 + \tfrac{3}{4}\varphi_5 \\
\varphi_2 &= \tfrac{1}{4}\varphi_1 + \tfrac{1}{4}\varphi_2 + 0\varphi_3 + \tfrac{1}{4}\varphi_4 + 0\varphi_5 \\
\varphi_3 &= 0\varphi_1 + \tfrac{1}{4}\varphi_2 + 0\varphi_3 + 0\varphi_4 + 0\varphi_5 \\
\varphi_4 &= 0\varphi_1 + 0\varphi_2 + \tfrac{1}{4}\varphi_3 + 0\varphi_4 + \tfrac{1}{4}\varphi_5 \\
\varphi_5 &= 0\varphi_1 + 0\varphi_2 + 0\varphi_3 + \tfrac{1}{4}\varphi_4 + 0\varphi_5
\end{aligned}$$

  and the total probability equation $\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4 + \varphi_5 = 1.$
- This is 6 equations in 5 unknowns, so it is overdetermined.
- Actually, the first 5 equations are underdetermined; they add up to

$$\varphi_1 + \cdots + \varphi_5 = \varphi_1 + \cdots + \varphi_5.$$

- Knock out the $\varphi_5 = \cdots$ equation and solve the rest of them to get $\vec{\varphi}\,' = (\frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256}) \approx (0.6875, 0.2344, 0.0586, 0.0156, 0.0039).$

Solving equations in Matlab or R

(this method doesn't use the functions for eigenvectors)

Matlab

```
>> eye(5) # identity
   1    0    0    0    0
   0    1    0    0    0
   0    0    1    0    0
   0    0    0    1    0
   0    0    0    0    1
>> P1' - eye(5) # transpose minus identity
 -0.2500  0.5000  0.7500  0.5000  0.7500
  0.2500 -0.7500       0  0.2500       0
       0  0.2500 -1.0000       0       0
       0       0  0.2500 -1.0000  0.2500
       0       0       0  0.2500 -1.0000
>> [P1' - eye(5) ; 1 1 1 1 1]
 -0.2500  0.5000  0.7500  0.5000  0.7500
  0.2500 -0.7500       0  0.2500       0
       0  0.2500 -1.0000       0       0
       0       0  0.2500 -1.0000  0.2500
       0       0       0  0.2500 -1.0000
  1.0000  1.0000  1.0000  1.0000  1.0000
>> sstate=[P1'-eye(5); 1 1 1 1 1] \ [0 0 0 0 0 1]'
sstate =
  0.6875
  0.2344
  0.0586
  0.0156
  0.0039
```

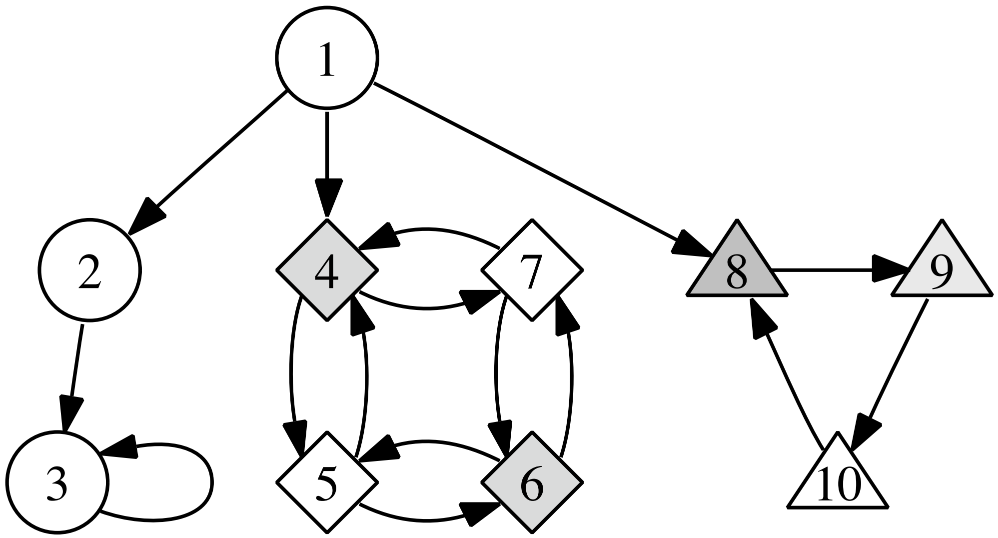
R

```
> diag(1,5) % identity
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
> t(P1) - diag(1,5) % transpose minus identity
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,] -0.25  0.50  0.75  0.50  0.75
[2,]  0.25 -0.75  0.00  0.25  0.00
[3,]  0.00  0.25 -1.00  0.00  0.00
[4,]  0.00  0.00  0.25 -1.00  0.25
[5,]  0.00  0.00  0.00  0.25 -1.00
> rbind(t(P1) - diag(1,5), c(1,1,1,1,1))
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,] -0.25  0.50  0.75  0.50  0.75
[2,]  0.25 -0.75  0.00  0.25  0.00
[3,]  0.00  0.25 -1.00  0.00  0.00
[4,]  0.00  0.00  0.25 -1.00  0.25
[5,]  0.00  0.00  0.00  0.25 -1.00
[6,]  1.00  1.00  1.00  1.00  1.00
> sstate = qr.solve(rbind(t(P1) - diag(1,5),
+     c(1,1,1,1,1)),c(0,0,0,0,0,1))
> sstate
[1] 0.68750000 0.23437500 0.05859375 0.01562500
[5] 0.00390625
```

Eigenvalues of P

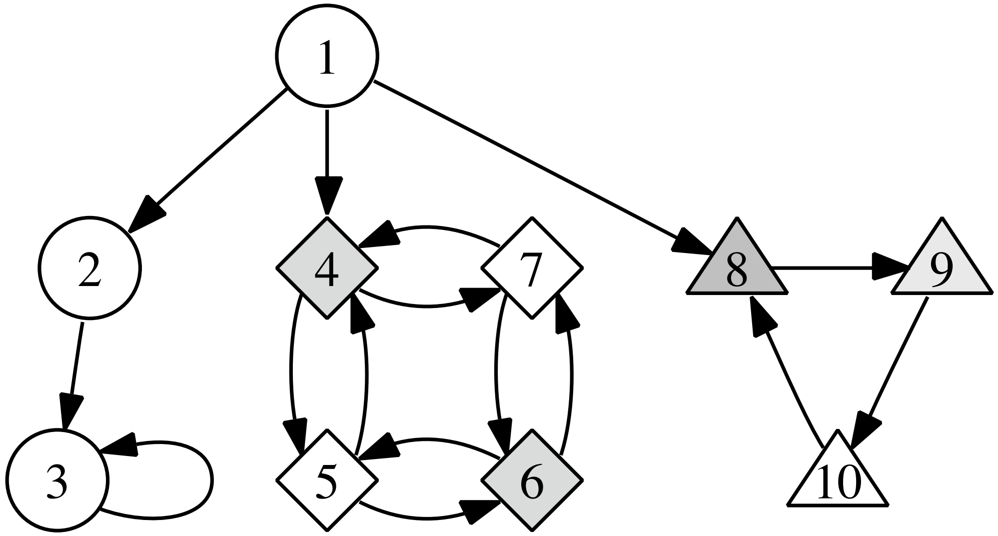
- A transition matrix is *stochastic*: all entries are $\geqslant 0$ and its row sums are all 1. So
$$P\vec{1} = \vec{1} \qquad \text{where } \vec{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$
- Thus, $\lambda = 1$ is an eigenvalue of P and $\vec{1}$ is a *right eigenvector*. There is also a *left eigenvector* of P with eigenvalue 1:
$$\vec{w}\,P = 1\vec{w}$$
where $\vec{w}$ is a row vector. Normalize it so its entries add up to 1, to get the *stationary distribution* $\vec{\varphi}\,'$.
- All eigenvalues λ of a stochastic matrix have $|\lambda| \leqslant 1$.
- An irreducible aperiodic Markov chain has just one eigenvalue $= 1$. The 2nd largest $|\lambda|$ determines how fast P^n converges. For example, if it's diagonalizable, the spectral decomposition is:
$$P^n = 1^n M_1 + {\lambda_2}^n M_2 + {\lambda_3}^n M_3 + \cdots$$
but there may be complications (periodic Markov chains, complex eigenvalues, . . .).

Technicalities — reducibility



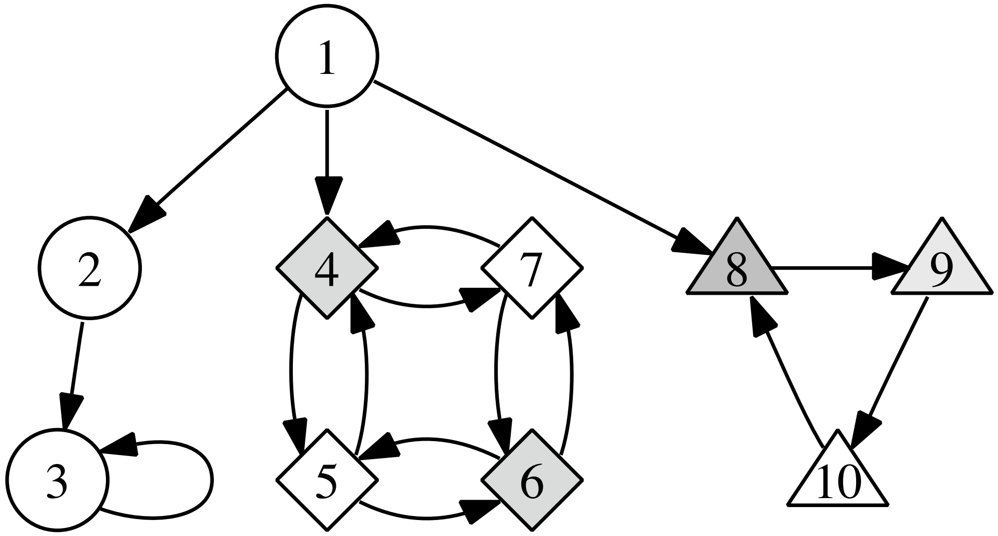
- A Markov chain is *irreducible* if every state can be reached from every other state after enough steps.
- The above example is *reducible* since there are states that cannot be reached from each other: after sufficient time, you are either stuck in state 3, the component $\{4, 5, 6, 7\}$, or the component $\{8, 9, 10\}$.

Technicalities — period



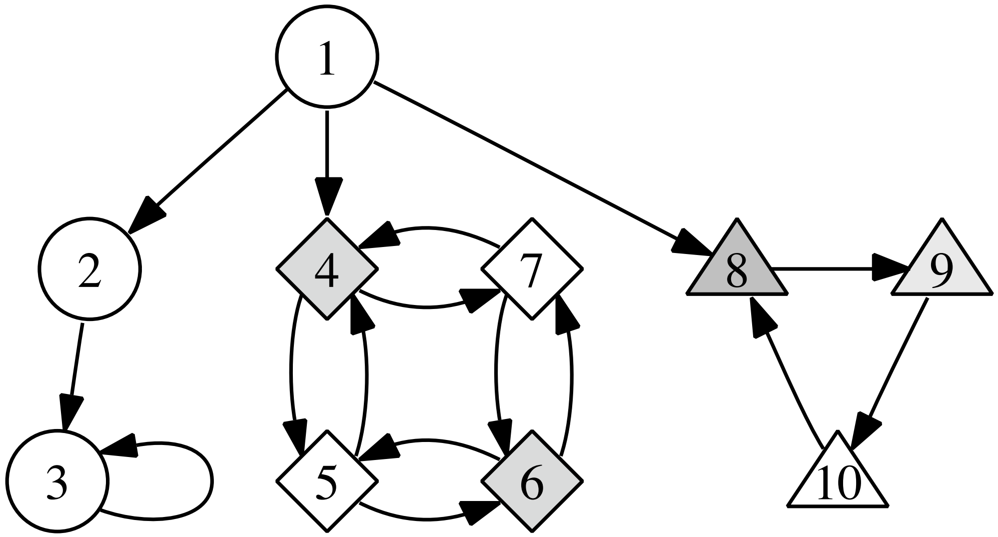
- State i has *period d* if the Markov chain can only go from state i to itself in multiples of d steps, where d is the maximum number that satisfies that.
- If $d > 1$ then state i is *periodic.*
- A Markov chain is *periodic* if at least one state is periodic and is *aperiodic* if no states are periodic.
- All states in a component have the same period.
- Component $\{4, 5, 6, 7\}$ has period 2 and component $\{8, 9, 10\}$ has period 3, so the Markov chain is periodic.

Technicalities — absorbing states



- An *absorbing state* has all its outgoing edges going to itself; e.g., state 3 above.
- An irreducible Markov chain with two or more states cannot have any absorbing states.

Technicalities — summary



- There are generalizations to infinite numbers of discrete or continuous states and to continuous time.
- We will work with Markov chains that are finite, discrete, irreducible, and aperiodic, unless otherwise stated.
- For a finite discrete Markov chain on two or more states:

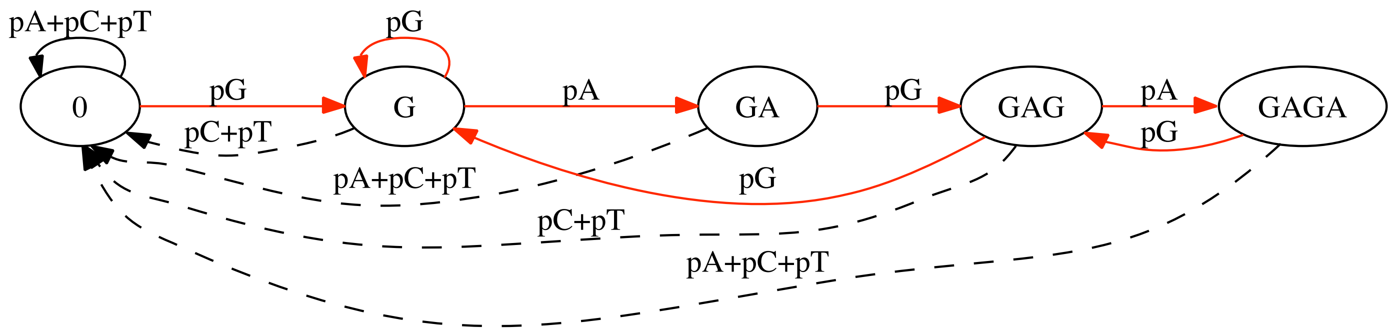
  irreducible and aperiodic with no absorbing states

  is equivalent to

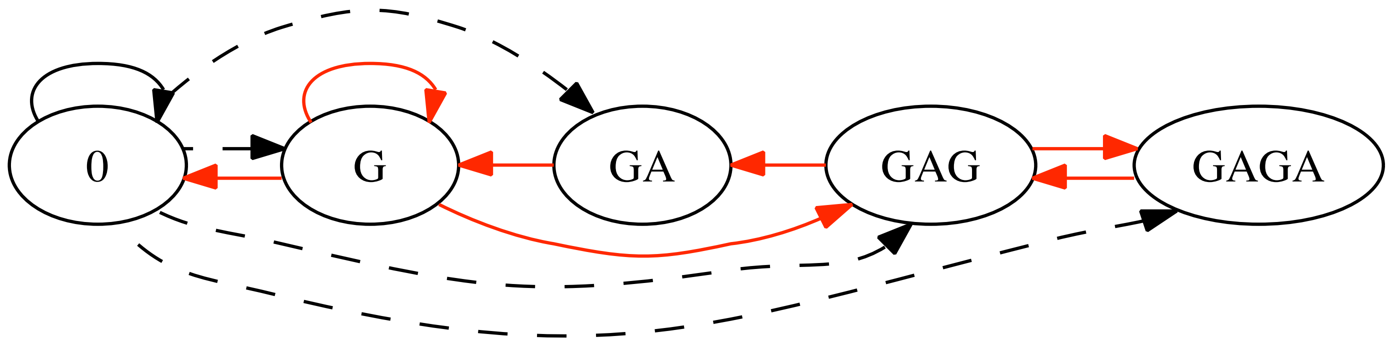
  P or a power of P has all entries greater than 0

  and in this case, $\lim_{n\to\infty} P^n$ exists and all its rows are the stationary distribution.

***M*1**



Reverse



- A Markov chain modeling forwards progression of time can be "reversed" to make "predictions" about the past. For example, this is done in models of nucleotide evolution.
- The graph of the reverse Markov chain has
  - the same nodes as the forwards chain;
  - the same edges but reversed directions and new probabilities.

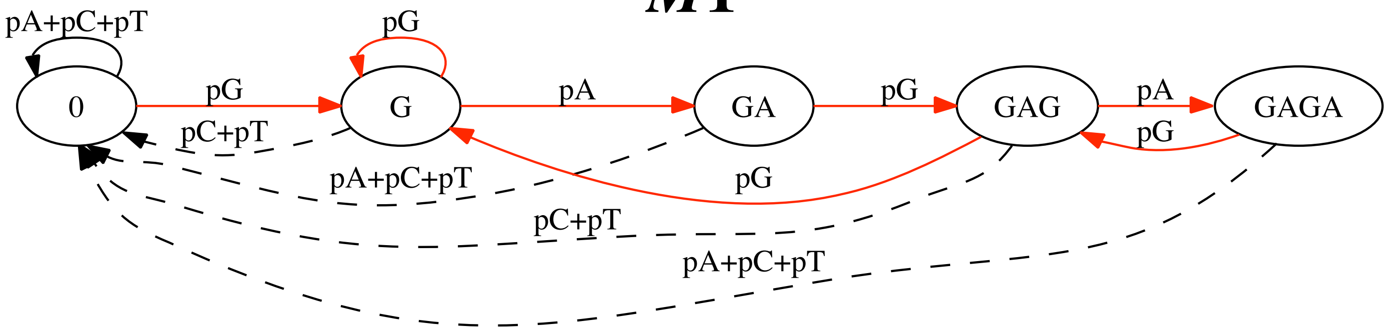
Reverse Markov Chain

- The transition matrix P of the forwards Markov chain was defined so that $P(X_{t+1} = j | X_t = i) = p_{ij}$ at all times t.
- Assume the forwards machine has run long enough to reach the stationary distribution, $P(X_t = i) = \varphi_i$.
- The reverse Markov chain has transition matrix Q, where

$$q_{ij} = P(X_t = j | X_{t+1} = i) = \frac{P(X_{t+1} = i | X_t = j) P(X_t = j)}{P(X_{t+1} = i)} = \frac{p_{ji} \varphi_j}{\varphi_i}$$

(Recall Bayes' Theorem: $P(B|A) = P(A|B)P(B)/P(A)$.)

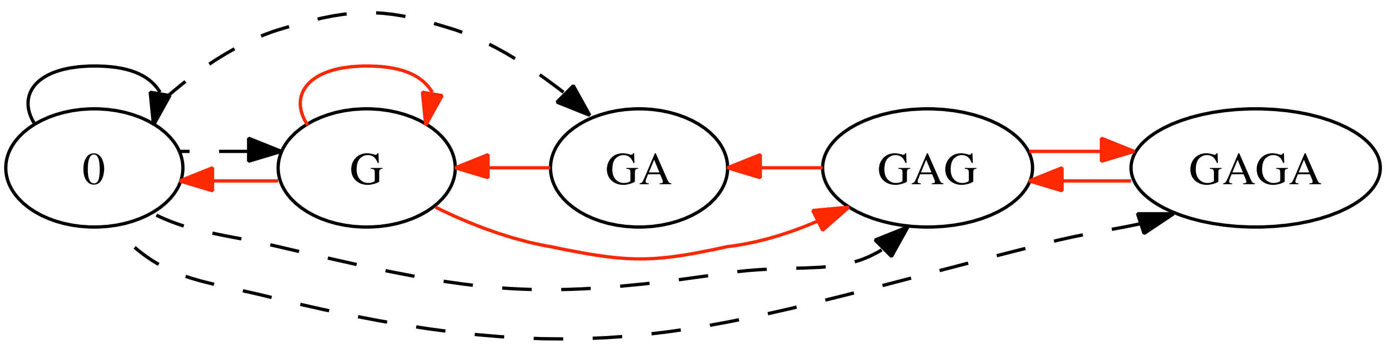
$M1$



$P1$

$$\begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \end{bmatrix}$$

Reverse of $M1$



$Q1$

$$\begin{bmatrix} \frac{3}{4} & \frac{15}{88} & \frac{45}{704} & \frac{1}{88} & \frac{3}{704} \\ \frac{11}{15} & \frac{1}{4} & 0 & \frac{1}{60} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{15}{16} & 0 & \frac{1}{16} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- Stationary distribution of $P1$ is $\vec{\varphi}' = \left(\frac{11}{16}, \frac{15}{64}, \frac{15}{256}, \frac{1}{64}, \frac{1}{256}\right)$
- **Example of one entry:** The edge `0` $\to$ `GA` in the reverse chain has $q_{13} = p_{31}\varphi_3/\varphi_1 = \left(\frac{3}{4}\right)\left(\frac{15}{256}\right)/\left(\frac{11}{16}\right) = \frac{45}{704}$.
- This means that in the steady state of the forwards chain, when `0` is entered, there is a probability $\frac{45}{704}$ that the previous state was `GA`.

Matlab and R

Matlab

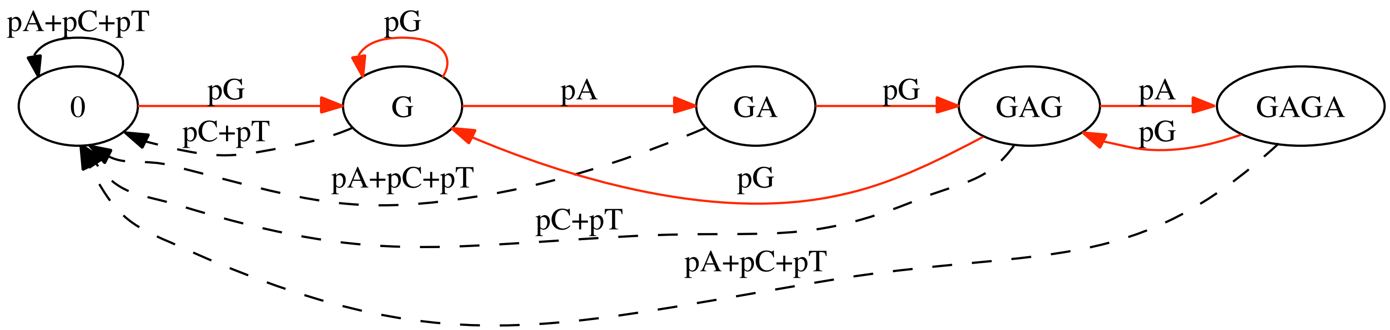
```
>> d_sstate = diag(sstate)
d_sstate =
    0.6875         0         0         0         0
         0    0.2344         0         0         0
         0         0    0.0586         0         0
         0         0         0    0.0156         0
         0         0         0         0    0.0039

>> Q1 = inv(d_sstate) * P1' * d_sstate
Q1 =
    0.7500    0.1705    0.0639    0.0114    0.0043
    0.7333    0.2500         0    0.0167         0
         0    1.0000         0         0         0
         0         0    0.9375         0    0.0625
         0         0         0    1.0000         0
```

R

```
> d_sstate = diag(sstate)
> Q1 = solve(d_sstate) %*% t(P1) %*% d_sstate
```

Expected time from state i till next time in state j

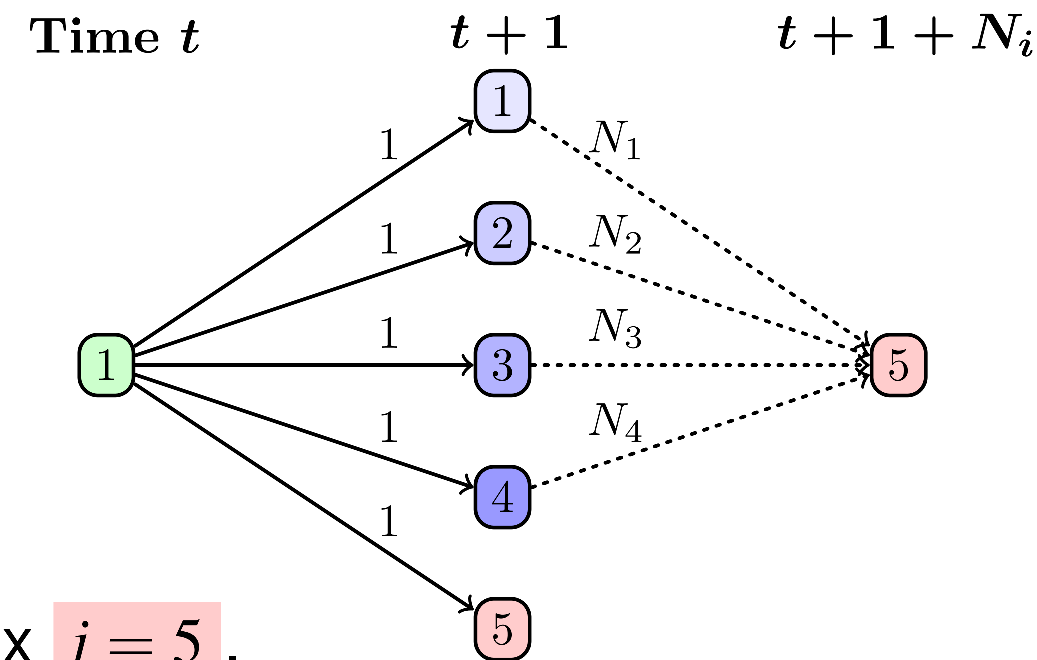


If $M1$ is in state $\emptyset$, what is the expected number of steps until the next time it is in state GAGA?

More generally, what's the expected # steps from state i to state j?

- Fix the end state j once and for all.
  Simultaneously solve for expected # steps from all start states i.
- For $i = 1, \ldots, s$, let N_i be a random variable for the number of steps from state i to the next time in state j.
- *Next time* means that if $i = j$, we count until the next time at state j, with $N_j \geqslant 1$; we don't count it as already there in 0 steps.
- We'll develop systems of equations for $E(N_i)$, $\mathrm{Var}(N_i)$, and $\mathbb{P}_{N_i}(x)$.

Expected time from state i till next time in state j



# Steps	Probability
N_1+1	P_{11}
N_2+1	P_{12}
N_3+1	P_{13}
N_4+1	P_{14}
1	P_{15}

- Fix $j=5$.
- Random variable N_r = # steps from state r to next time in state j.
- Dotted paths have no occurrences of state j in the middle.
- Expected # steps from state $i=1$ to $j=5$ (repeat this for all i):

$$E(N_1^{\text{(time } t)}) = P_{11}\,E(N_1^{\text{(time } t+1)}+1) + P_{12}\,E(N_2+1)$$
$$+ P_{13}\,E(N_3+1) + P_{14}\,E(N_4+1) + P_{15}\,E(1)$$

Both N_1's have same distribution, and we can expand $E()$'s:

$$E(N_1) = \sum_{r:r\neq j} P_{1r}\,E(N_r) + \sum_{r} P_{1r} = \Big(\sum_{r:r\neq j} P_{1r}\,E(N_r)\Big) + 1$$

Expected time from state i till next time in state j

- Recall we fixed j, and defined N_i relative to it.
- Start in state i.
- There is a probability P_{ir} of going one step to state r.
- If $r = j$, we are done in one step: $E(N_i \,|\, \text{1st step is } i \to j) = 1$
  If $r \neq j$, the expected number of steps after the first step is $E(N_r)$:
  $$E(N_i \,|\, \text{1st step is } i \to r) = E(N_r + 1) = E(N_r) + 1$$
- Combine with the probability of each value of r:

$$\begin{aligned} E(N_i) &= P_{ij} \cdot 1 + \sum_{r=1,\, r\neq j}^{s} P_{ir} E(N_r + 1) = P_{ij} + \sum_{r=1,\, r\neq j}^{s} P_{ir} \cdot (E(N_r) + 1) \\ &= \sum_{r=1}^{s} P_{ir} + \sum_{r=1,\, r\neq j}^{s} P_{ir} \cdot E(N_r) = 1 + \sum_{r=1,\, r\neq j}^{s} P_{ir} \cdot E(N_r) \end{aligned}$$

- Doing this for all s states, $i = 1, \ldots, s$, gives s equations in the s unknowns $E(N_1), \ldots, E(N_s)$.

Expected times between states in $M1$: times to state 5

$$E(N_1) = 0 + \tfrac{3}{4}(E(N_1)+1) + \tfrac{1}{4}(E(N_2)+1) = 1 + \tfrac{3}{4}E(N_1) + \tfrac{1}{4}E(N_2)$$
$$E(N_2) = 0 + \tfrac{1}{2}(E(N_1)+1) + \tfrac{1}{4}(E(N_2)+1) + \tfrac{1}{4}(E(N_3)+1) = 1 + \tfrac{1}{2}E(N_1) + \tfrac{1}{4}E(N_2) + \tfrac{1}{4}E(N_3)$$
$$E(N_3) = 0 + \tfrac{3}{4}(E(N_1)+1) + \tfrac{1}{4}(E(N_4)+1) = 1 + \tfrac{3}{4}E(N_1) + \tfrac{1}{4}E(N_4)$$
$$E(N_4) = \tfrac{1}{4} + \tfrac{1}{2}(E(N_1)+1) + \tfrac{1}{4}(E(N_2)+1) = 1 + \tfrac{1}{2}E(N_1) + \tfrac{1}{4}E(N_2)$$
$$E(N_5) = 0 + \tfrac{3}{4}(E(N_1)+1) + \tfrac{1}{4}(E(N_4)+1) = 1 + \tfrac{3}{4}E(N_1) + \tfrac{1}{4}E(N_4)$$

- This is 5 equations in 5 unknowns $E(N_1),\ldots,E(N_5)$. Matrix format:

$$\underbrace{\begin{bmatrix} -1/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & -3/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & -1 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & -1 & 0 \\ 3/4 & 0 & 0 & 1/4 & -1 \end{bmatrix}}_{\substack{\text{Zero out } j^{\text{th}} \text{ column of } P. \\ \text{Then subtract 1 from each diagonal entry.}}} \begin{bmatrix} E(N_1) \\ E(N_2) \\ E(N_3) \\ E(N_4) \\ E(N_5) \end{bmatrix} = \underbrace{\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}}_{\text{All } -1\text{'s.}}$$

- $E(N_1) = 272$, $E(N_2) = 268$, $E(N_3) = 256$, $E(N_4) = 204$, $E(N_5) = 256$.
- **Matlab and R:** Enter matrix `C` and vector `r`. Solve $C\vec{x} = \vec{r}$ with
  Matlab: `x=C\r` or `x=inv(C)*r` **R:** `x=solve(C,r)`

Variance and PGF of number of steps between states

- We may compute $E(g(N_i))$ for any function g by setting up recurrences in the same way. For each $i = 1, \ldots, s$:
$$E(g(N_i)) = P_{ij}\, g(1) + \sum_{r \neq j} P_{ir}\, E(g(N_r+1)) = \text{expansion depending on } g$$

- **Variance of N_i's:** $\mathrm{Var}(N_i) = E({N_i}^2) - (E(N_i))^2$
$$E({N_i}^2) = P_{ij} \cdot 1^2 + \sum_{r=1,\ r \neq j}^{s} P_{ir}\, E((N_r+1)^2) = 1 + 2 \sum_{r=1,\ r \neq j}^{s} P_{ir}\, E(N_r) + \sum_{r=1,\ r \neq j}^{s} P_{ir}\, E({N_r}^2)$$
Plug in the previous solution of $E(N_1), \ldots, E(N_s)$.
Then solve the s equations for the s unknowns $E({N_1}^2), \ldots, E({N_s}^2)$.

- **PGF:** $\mathbb{P}_{N_i}(x) = E(x^{N_i}) = \sum_{n=0}^{\infty} P(N_i = n) x^n$
$$E(x^{N_i}) \;=\; P_{ij} \cdot x^1 + \sum_{r=1,\ r \neq j}^{s} P_{ir}\, E(x^{N_r+1}) \;=\; P_{ij} \cdot x + \sum_{r=1,\ r \neq j}^{s} P_{ir} \cdot x \cdot E(x^{N_r})$$
Solve the s equations for s unknowns $E(x^{N_1}), \ldots, E(x^{N_s})$.
See the old handout for a worked out example.

Powers of matrices (see separate slides)

- **Sample matrix:** **Diagonalization:** $P = VDV^{-1}$

  $P = \begin{bmatrix} 8 & -1 \\ 6 & 3 \end{bmatrix}$ $V = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $D = \begin{bmatrix} 5 & 0 \\ 0 & 6 \end{bmatrix}$ $V^{-1} = \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$

- $P^n = (VDV^{-1})(VDV^{-1}) \cdots (VDV^{-1}) = VD^nV^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5^n & 0 \\ 0 & 6^n \end{bmatrix} \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$

- When a square ($s \times s$) matrix P has distinct eigenvalues, it can be *diagonalized*

  $$P = VDV^{-1}$$

  where D is a diagonal matrix of the eigenvalues of P (any order); the columns of V are right eigenvectors of P (in same order as D); the rows of V^{-1} are left eigenvectors of P (in same order as D);

- If any eigenvalues are equal, it may or may not be diagonalizeable, but there is a generalization called *Jordan Canonical Form*, $P = VJV^{-1}$ giving $P^n = VJ^nV^{-1}$.
  J has eigenvalues on the diagonal and 1's and 0's just above it, and is also easy to raise to powers.

Matrix powers — spectral decomposition (distinct eigenvalues)

- **Powers of P:** $P^n = (VDV^{-1})(VDV^{-1})\cdots = VD^nV^{-1}$

$$P^n = VD^nV^{-1} \;=\; V\begin{bmatrix}5^n & 0\\ 0 & 6^n\end{bmatrix}V^{-1} = V\begin{bmatrix}5^n & 0\\ 0 & 0\end{bmatrix}V^{-1} + V\begin{bmatrix}0 & 0\\ 0 & 6^n\end{bmatrix}V^{-1}$$

$$\begin{aligned}
V\begin{bmatrix}5^n & 0\\ 0 & 0\end{bmatrix}V^{-1} &= \begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}5^n & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}-2 & 1\\ 1.5 & -.5\end{bmatrix} = \begin{bmatrix}(1)(5^n)(-2) & (1)(5^n)(1)\\ (3)(5^n)(-2) & (3)(5^n)(1)\end{bmatrix}\\
&= 5^n\begin{bmatrix}1\\ 3\end{bmatrix}\begin{bmatrix}-2 & 1\end{bmatrix} = {\lambda_1}^n\vec{r}_1\vec{\ell}_1' = 5^n\begin{bmatrix}-2 & 1\\ -6 & 3\end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
V\begin{bmatrix}0 & 0\\ 0 & 6^n\end{bmatrix}V^{-1} &= \begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 6^n\end{bmatrix}\begin{bmatrix}-2 & 1\\ 1.5 & -.5\end{bmatrix} = \begin{bmatrix}2(6^n)(1.5) & 2(6^n)(-.5)\\ 4(6^n)(1.5) & 4(6^n)(-.5)\end{bmatrix}\\
&= 6^n\begin{bmatrix}2\\ 4\end{bmatrix}\begin{bmatrix}1.5 & -.5\end{bmatrix} = {\lambda_2}^n\vec{r}_2\vec{\ell}_2' = 6^n\begin{bmatrix}3 & -1\\ 6 & -2\end{bmatrix}
\end{aligned}$$

- **Spectral decomposition of P^n:**

$$P^n = VD^nV^{-1} = {\lambda_1}^n\vec{r}_1\vec{\ell}_1' + {\lambda_2}^n\vec{r}_2\vec{\ell}_2' = 5^n\begin{bmatrix}-2 & 1\\ -6 & 3\end{bmatrix} + 6^n\begin{bmatrix}3 & -1\\ 6 & -2\end{bmatrix}$$

Jordan Canonical Form

- Matrices with two or more equal eigenvalues cannot necessarily be diagonalized. *Matlab and R do not give an error or warning.*
- The *Jordan Canonical Form* is a generalization that turns into diagonalization when possible, and still works otherwise:

$$P = VJV^{-1} \qquad J = \begin{bmatrix} B_1 & 0 & 0 & \cdots \\ 0 & B_2 & 0 & \cdots \\ 0 & 0 & B_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad B_i = \begin{bmatrix} \lambda_i & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_i & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda_i & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \lambda_i \end{bmatrix}$$

- $P^n = VJ^nV^{-1}$ where

$$J^n = \begin{bmatrix} {B_1}^n & 0 & 0 & \cdots \\ 0 & {B_2}^n & 0 & \cdots \\ 0 & 0 & {B_3}^n & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad {B_i}^n = \begin{bmatrix} {\lambda_i}^n & \binom{n}{1}{\lambda_i}^{n-1} & \binom{n}{2}{\lambda_i}^{n-2} & \cdots\cdots & \cdots \\ 0 & {\lambda_i}^n & \binom{n}{1}{\lambda_i}^{n-1} & \cdots\cdots & \cdots \\ \ddots & \ddots & \ddots & \ddots\ \ddots & \ddots \\ 0 & 0 & 0 & \cdots\ {\lambda_i}^n & \binom{n}{1}{\lambda_i}^{n-1} \\ 0 & 0 & 0 & \cdots\ 0 & {\lambda_i}^n \end{bmatrix}$$

- In applications when repeated eigenvalues are a possibility, it's best to use the Jordan Canonical Form.

Jordan Canonical Form for $P1$ in Matlab

(R doesn't currently have JCF available either built-in or as an add-on)

```
» P1 = [
 [ 3/4, 1/4,    0,    0,    0 ];  %
 [ 2/4, 1/4, 1/4,    0,    0 ];  % G
 [ 3/4,    0,    0, 1/4,    0 ];  % GA
 [ 2/4, 1/4,    0,    0, 1/4 ];  % GAG
 [ 3/4,    0,    0, 1/4,    0 ];  % GAGA
]

P1 =
   0.7500   0.2500        0        0        0
   0.5000   0.2500   0.2500        0        0
   0.7500        0        0   0.2500        0
   0.5000   0.2500        0        0   0.2500
   0.7500        0        0   0.2500        0

» [V1,J1] = jordan(P1)
V1 =
  -0.0039  -0.0195  -0.0707  -0.2298  -0.0430
   0.0117   0.0430   0.1339   0.4066  -0.0430
  -0.0039   0.0430   0.1793   0.5884  -0.0430
   0.0117   0.0430   0.3839   1.4066  -0.0430
  -0.0039   0.0430   0.1793   1.5884  -0.0430

J1 =
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     0
     0     0     0     0     1
```

```
» V1i = inv(V1)
V1i =
         0   52.3636  -64.0000   11.6364         0
  -16.0000   16.0000   13.0909  -16.0000    2.9091
         0   -4.0000    4.0000    4.0000   -4.0000
         0         0   -1.0000         0    1.0000
  -16.0000   -5.4545   -1.3636   -0.3636   -0.0909

» V1 * J1 * V1i
ans =
    0.7500    0.2500   -0.0000   -0.0000    0.0000
    0.5000    0.2500    0.2500   -0.0000    0.0000
    0.7500   -0.0000   -0.0000    0.2500   -0.0000
    0.5000    0.2500    0.0000   -0.0000    0.2500
    0.7500   -0.0000   -0.0000    0.2500   -0.0000
```

Powers of $P1$ using JCF

- $P = VJV^{-1}$ gives $P^n = VJ^nV^{-1}$, and J^n is easy to compute:

```
» J1
J1 =
   0   1   0   0   0
   0   0   1   0   0
   0   0   0   1   0
   0   0   0   0   0
   0   0   0   0   1
```

```
» J1^2
ans =
   0   0   1   0   0
   0   0   0   1   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   1
```

```
» J1^3
ans =
   0   0   0   1   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   1
```

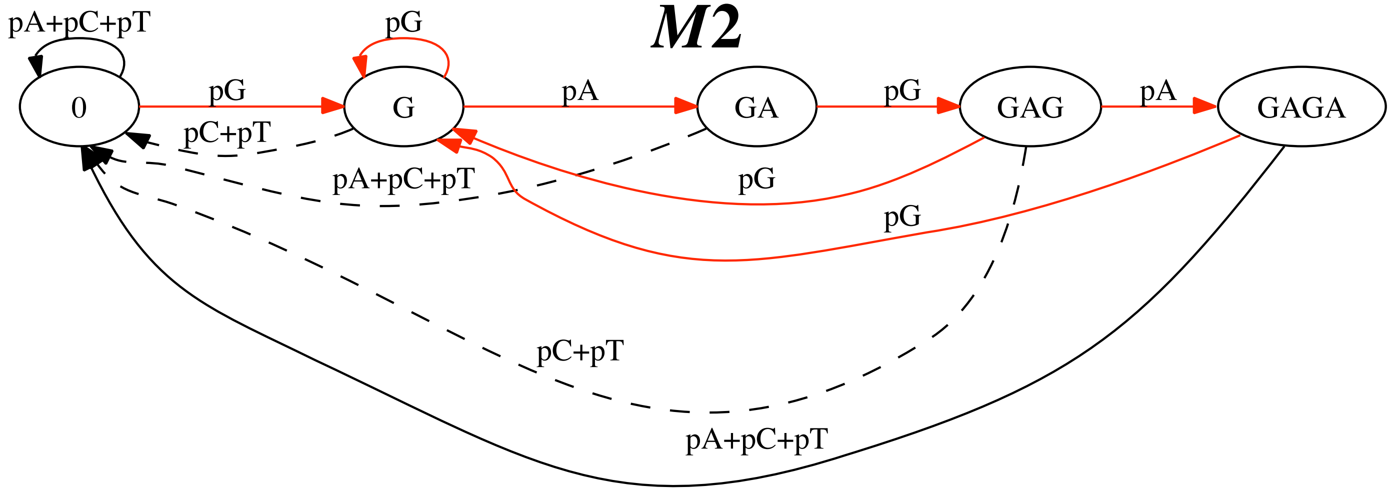
```
» J1^4
ans =
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   1
```

```
» J1^5
ans =
   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
```

- For this matrix, $J^n = J^4$ when $n \geqslant 4$, so
  $P^n = VJ^nV^{-1} = VJ^4V^{-1} = P^4$ for $n \geqslant 4$.

Non-overlapping occurrences of GAGA

M2



P2

$$\begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 0 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 0 & 1/4 \\ 3/4 & 1/4 & 0 & 0 & 0 \end{bmatrix}$$

```
» [V2,J2] = jordan(P2)
V2 =
  -0.0625          -0.5170          -0.1728           0.1176 + 0.0294i   0.1176 - 0.0294i
   0.1875           1.3011          -0.1728          -0.3824 + 0.0294i  -0.3824 - 0.0294i
  -0.0625           0.4830          -0.1728           0.1176 - 0.4706i   0.1176 + 0.4706i
   0.1875           1.3011          -0.1728           0.1176 + 0.0294i   0.1176 - 0.0294i
  -0.0625           0.4830          -0.1728           0.1176 + 0.0294i   0.1176 - 0.0294i

J2 =     0           1.0000                0                0                  0
         0                0                0                0                  0
         0                0           1.0000                0                  0
         0                0                0                0 + 0.2500i        0
         0                0                0                0                  0 - 0.2500i

» V2i = inv(V2)
V2i =
   3.2727                0          -0.0000           4.0000            -7.2727
  -1.0000                0           0.0000                0             1.0000
  -3.9787          -1.3617          -0.3404          -0.0851            -0.0213
        0          -1.0000                0 + 1.0000i   1.0000                 0 - 1.0000i
        0          -1.0000                0 - 1.0000i   1.0000                 0 + 1.0000i
```

Non-overlapping occurrences of GAGA — JCF

$$(J2)^n = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n & & & \\ & 1^n & & \\ & & (i/4)^n & \\ & & & (-i/4)^n \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ for } n \geqslant 2$$

- **One eigenvalue = 1.** It's the third one listed, so the stationary distribution is the third row of $(V2)^{-1}$ normalized:

```
» V2i(3,:) / sum(V2i(3,:))
ans =
    0.6875    0.2353    0.0588    0.0147    0.0037
```

- **Two eigenvalues = 0.** The interpretation of one of them is that the first and last rows of $P2$ are equal, so $(1, 0, 0, 0, -1)'$ is a right eigenvector of $P2$ with eigenvalue 0.
- **Two complex eigenvalues, $\mathbf{0 \pm i/4}$.** Since $P2$ is real, all complex eigenvalues must come in conjugate pairs.
  The eigenvectors also come in conjugate pairs (last 2 columns of $V2$; last 2 rows of $(V2)^{-1}$.

Spectral decomposition with JCF and complex eigenvalues

$$(J2)^n = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n & & & \\ & 1^n & & \\ & & (i/4)^n & \\ & & & (-i/4)^n \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ for } n \geqslant 2$$

$$\begin{aligned} (\mathtt{P2})^n &= (\mathtt{V2})(\mathtt{J2})^n(\mathtt{V2})^{-1} \\ &= \begin{bmatrix} \vec{r}_1 & \vec{r}_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^n \begin{bmatrix} \vec{\ell}'_1 \\ \vec{\ell}'_2 \end{bmatrix} + \vec{r}_3 (1)^n \vec{\ell}'_3 + \vec{r}_4 (i/4)^n \vec{\ell}'_4 + \vec{r}_5 (-i/4)^n \vec{\ell}'_5 \end{aligned}$$

The first term vanishes when $n \geqslant 2$, so when $n \geqslant 2$ the format is

$$= 1^n \mathtt{S3} + (i/4)^n \mathtt{S4} + (-i/4)^n \mathtt{S5} = \mathtt{S3} + (i/4)^n \mathtt{S4} + (-i/4)^n \mathtt{S5}$$

Spectral decomposition with JCF and complex eigenvalues

For $n \geqslant 2$, $\quad (\texttt{P2})^n = \texttt{S3} + (i/4)^n\texttt{S4} + (-i/4)^n\texttt{S5} \quad$ where

```
» S3 = V2(:,3) * V2i(3,:)
S3 =
    0.6875    0.2353    0.0588    0.0147    0.0037
    0.6875    0.2353    0.0588    0.0147    0.0037
    0.6875    0.2353    0.0588    0.0147    0.0037
    0.6875    0.2353    0.0588    0.0147    0.0037
    0.6875    0.2353    0.0588    0.0147    0.0037

» S4 = V2(:,4) * V2i(4,:)
S4 =     0             -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i
         0              0.3824 - 0.0294i  -0.0294 - 0.3824i  -0.3824 + 0.0294i   0.0294 + 0.3824i
         0             -0.1176 + 0.4706i   0.4706 + 0.1176i   0.1176 - 0.4706i  -0.4706 - 0.1176i
         0             -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i
         0             -0.1176 - 0.0294i  -0.0294 + 0.1176i   0.1176 + 0.0294i   0.0294 - 0.1176i

» S5 = V2(:,5) * V2i(5,:)
S5 =     0             -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
         0              0.3824 + 0.0294i  -0.0294 + 0.3824i  -0.3824 - 0.0294i   0.0294 - 0.3824i
         0             -0.1176 - 0.4706i   0.4706 - 0.1176i   0.1176 + 0.4706i  -0.4706 + 0.1176i
         0             -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
         0             -0.1176 + 0.0294i  -0.0294 - 0.1176i   0.1176 - 0.0294i   0.0294 + 0.1176i
```

- `S3` corresponds to the stationary distribution.
- `S4` and `S5` are complex conjugates, so $(i/4)^n\texttt{S4} + (-i/4)^n\texttt{S5}$ is a sum of two complex conjugates; thus, it is real-valued, even though complex numbers are involved in the computation.