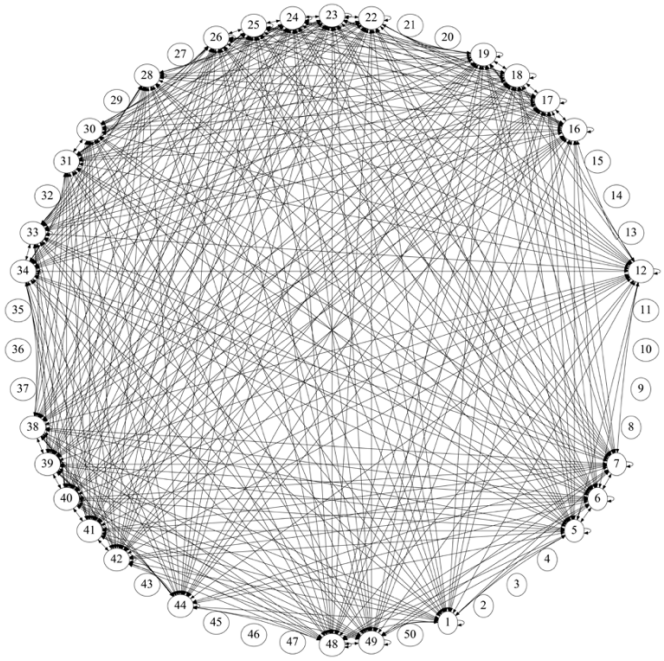
BIM 105
Fall 2024

Lab 8: Modeling Cancer with Markov Chains

Instructions: Read the prompts and use MATLAB to answer the questions. Submit your solution as a single script which outputs all answers to the Command Window. A template script is available on Canvas.



Part 1. Progression of Metastatic Cancer

Metastasis is the medical term for cancer that spreads to a different part of the body from where it originally started (the primary site). Cancer cells which enter the circulatory system and disperse through the body are called “circulating cancer cells” (CTCs). Treatment of metastatic cancer is particularly challenging, and as a consequence the movement of CTCs in the human body has been of significant scientific research. A better understanding of how CTCs disperse from one area of the body to another could help medical professionals predict the progression of metastatic cancer and ultimately improve potential therapeutics.

A paper published by Newton et al[1] in 2012 modeled the movement of cancer cells through the body as a Markov process. In their study, they compiled data from over 3000 autopsies to characterize the propensity of metastatic cancer to migrate from various “primary sites” to other organs in the body. They considered 50 distinct locations in the body as the states for their Markov model and used an optimization algorithm to convert the autopsy data into model parameters.

[1] Newton PK, Mason J, Bethel K, Bazhenova LA, Nieva J, et al. (2012) A Stochastic Markov Chain Model to Describe Lung Cancer Growth and Metastasis. PLOS ONE 7(4): e34637. https://doi.org/10.1371/journal.pone.0034637

The result of their work is a 50-by-50 transition matrix encoding the probability of metastatic cancer to travel from one organ to another (the list of organs with their corresponding index for this matrix is provided on the last page of this lab). The transition matrix from the study is available on Canvas as "**`cancer_model.xlsx`**".

A graphical representation of a Markov system can be helpful to visualize how its states are connected. Newton et al provide such an illustration for their metastatic cancer model, and it's shown in this lab above the Part 1 header. Note that edges in this graph are shown only if the two states communicate with each other; in other words, $P_{ij} > 0$ and $P_{ji} > 0$.

With so many states and connections in the system, it becomes difficult to make any significant interpretation from the graph alone. Nevertheless, there is some information to be gleaned. Note that 23 of the 50 states do not communicate with any states (and therefore don't have any edges drawn in the graph). More specifically, 23 of the sites considered in the study were not observed to be colonized by metastatic cancer cells. Keep in mind that they could still be a primary cancer site that metastasizes to other organs, meaning that cancer in these sites can transition to other sites, but not vice versa.

Question 1: What is the probability to transition to any the 23 sites mentioned above? (in other words, what should the columns corresponding to those sites look like?)

Question 2: Load the transition matrix into a MATLAB script with the command **`P = xlsread('cancer_model.xls')`**. Display one of the columns that corresponds to a site not connected in the graph above and confirm that it agrees with your answer from Question 1.

Specifically, the Newton et al study was focused on the progression of metastatic lung cancer. Referring to the site-index table at the end of this lab, we know that the 23rd row of the transition matrix (**`P(23,:)`**) tells us the probabilities of cancer moving from the lungs to each the other sites.

Question 3: Plot this probability distribution using the **`bar`** command. Of the 23 sites that lung cancer can migrate to (including "self-seeding" back on itself), which site is most likely?

Question 4: Compute the limiting distribution of metastasis if the primary site is the lungs.

We can use the transition matrix of the model to simulate individual state sequences in MATLAB. This amounts to selecting an initial state (e.g., the lungs), sampling the next state according the probabilities in the transition matrix, and repeating this process a number of times. A function that does this for you is posted on Canvas as **`sim_MC.m`**.

To simulate a trajectory of cancer progression, you can call **`sim_MC(P, X0, N)`**. **`P`** is the transition matrix of the system; **`X0`** is the initial state, and **`N`** is the length of the simulated chain. For example, if you wanted to simulate a chain of length 100 starting with the lungs, you would call:

```
rng(107); % Sets the random seed for reproducibility
P = xlsread('cancer_model.xls')
chain = sim_MC(P, 23, 100);
```

Which produces the following chain:

```
23   6   25   24   17   22   6   22   ...
```

Question 5: Use the **`sim_MC`** function to generate some trajectories of cancer progression from the lung. Qualitatively, does the body site from your answer to Question 3 tend to appear at a high frequency?

The authors of the Newton et al study used simulations like this to estimate the **mean first-passage time** of sites in the body. The first-passage time of a site is the number of time steps required before the cancer has transitioned to that site. For example, in the simulated chain above Question 5, the first passage time of site 17 (the heart) was 4. The first-passage time for a site, however, will differ across simulations due to the random nature of the system (first-passage time is itself a random variable). Therefore, it's desirable to compute a *mean* first-passage time to better characterize the tendency of cancer to spread to certain sites.

Let's use simulations to estimate the mean first-passage time of some organs. To do this, we will need to run many simulations due to the relatively larger number of sites. The length of each simulation also needs to be sufficiently long, such that sites are reached at least one time. Given a simulated trajectory, we also need a way to compute the first-passage time of a site programmatically. We can accomplish this with the MATLAB command **`find`**, which returns where a value occurs in a given vector. For example, if we wanted to compute the first-passage time of the heart, we could use **`find(chain==17, 1)-1`**. Note that the second argument tells the function *how many* locations we want to find (we only care about the first location), and we also need to subtract 1, because the first element of the chain is **`X0`** (where no transitions have occurred yet).

Question 6: Generate 1000 simulations starting from the lungs with length 1000.

a. Use the trajectories to compute the mean first-passage time to the heart.
b. Use the trajectories to compute the mean first-passage time to the pancreas.

c. Use the trajectories to compute the mean first-passage time to the body site from your answer to Question 3. Is the mean first-passage time of this site lower than your answers to 6a and 6b?

Simulations allow us to estimate the mean first passage time based on the law of large numbers. Ideally, however, it would be better if we had a way to compute the mean first passage time to an organ without having to "brute force" an approximation through a large number of simulations. To do this, we can pose the question as a linear algebra problem, where the solution is the mean first passage time. By conditional probabilities, the expected time it takes to reach site i from site k can be written in terms of the expected time it takes to reach site i from all other sites and the transition probabilities to those sites. More formally:

$$T_k^i = 1 + \sum_{j \epsilon m} P_{kj} T_j^i$$

where m is the subset of states excluding i. Our objective is to solve for T^i, a vector of length equal to the number of sites in m. Using some matrix manipulation, we can arrive at the solution

$$T^i = (\boldsymbol{I} - \boldsymbol{Q})^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{Q}$ is the submatrix of transitions for states in m. The result of this computation gives us a vector containing the mean first-passage time to organ i given that the system started in organ k.

Let's do an example to see this solution in practice. Suppose that we are interested in the mean first-passage time to the heart, assuming the initial state is the lungs. First, we need to compose a *submatrix* of transitions for all states except the heart. In MATLAB, we can say:

```
Q = P;
Q(17,:) = 0;
Q(:,17) = 0;
```

Setting the corresponding elements of the transition matrix to zero has the same effect as constructing a new submatrix with those rows and columns omitted. Next, we need to solve the equation above, which amounts to:

```
T = inv((eye(50)-Q))*ones(50,1);
```
or
```
T = (eye(50)-Q)\ones(50,1);
```

Both of these statements will give the same solution, but often times it is faster and more reliable to use the back-slash operator in MATLAB ('\') instead of the **`inv`** command. The result, **`T`**, is a vector that tells us the expected time to reach the heart from each of the other organs. Thus, **`T(23)`** contains the mean first-passage time to the heart from the lungs, and is equal to **`T(23)=36.05`**.

Question 7: Use this approach to determine the mean first-passage times from the lungs to the pancreas and to the body site from your answer to Question 3 (assuming the initial state is the lungs). How do the results compare to the results from your simulations in Question 6?

#	Name	#	Name
1	Adrenal*	26	Omentum*
2	Anus	27	Ovaries
3	Appendix	28	Pancreas*
4	Bile Duct	29	Penis
5	Bladder	30	Pericardium*
6	Bone*	31	Peritoneum*
7	Brain*	32	Pharynx
8	Branchial Cyst	33	Pleura*
9	Breast	34	Prostate*
10	Cervix	35	Rectum
11	Colon	36	Retroperitoneum
12	Diaphragm*	37	Salivary
13	Duodenum	38	Skeletal Muscle*
14	Esophagus	39	Skin*
15	Eye	40	Small Intestine*
16	Gallbladder*	41	Spleen*
17	Heart*	42	Stomach*
18	Kidney*	43	Testes
19	Large Intestine*	44	Thyroid*
20	Larynx	45	Tongue
21	Lip*	46	Tonsil
22	Liver*	47	Unknown
23	Lung*	48	Uterus*
24	Lymph Nodes (reg)*	49	Vagina*
25	Lymph Nodes (dist)*	50	Vulva

Site numbering system used in transition matrix and network model. The * indicates an entry in the target vector associated with lung cancer primary from the data set of [6].

doi:10.1371/journal.pone.0034637.t001