Trabalho Prático 03

Luciano Stork

7 de outubro de 2023

Resumo

A análise de processos estocásticos desempenha um papel fundamental em diversas áreas da ciência e engenharia, oferecendo uma abordagem valiosa para modelar sistemas complexos e entender o comportamento de sistemas que evoluem com algum grau de aleatoriedade. Nesse contexto, este trabalho acadêmico se propõe a explorar os conceitos de processos estocásticos, taxa de entropia e cadeias de Markov por meio da utilização do software MATLAB. A tarefa em questão se desdobra em duas partes essenciais: a análise de uma cadeia de Markov utilizando dados reais de sequências de DNA, o que permite uma aplicação prática desses conceitos em um contexto biológico; e o cálculo da taxa de entropia da matriz de transição de uma cadeia de Markov, o que contribui para a compreensão das propriedades da informação e sua quantificação em sistemas diversos. Este trabalho busca não apenas aprofundar o conhecimento teórico desses conceitos, mas também desenvolver habilidades práticas de análise e modelagem em um ambiente computacional amplamente utilizado na pesquisa científica e na indústria.

1 Comentários Introdutórios

Avançando um pouco mais na disciplina, foi proposta uma tarefa que nos conduz a uma exploração profunda de conceitos cruciais em teoria da informação e processos estocásticos. Agora, estamos imersos na análise detalhada de cadeias de Markov, que desempenham um papel essencial na modelagem de sistemas dinâmicos. A compreensão de como os estados de um sistema evoluem ao longo do tempo, por meio da análise de cadeias de Markov, tem aplicações que variam desde previsões meteorológicas até a compreensão de processos biológicos complexos. Além disso, estamos investigando a taxa de entropia, uma medida fundamental de incerteza e informação em sistemas. Nossa análise desses conceitos é enriquecida pela aplicação a dados reais de sequências de DNA, permitindo-nos aplicar nosso conhecimento teórico em cenários do mundo real. Isso nos oferece a oportunidade de obter insights significativos sobre a natureza e complexidade das sequências genômicas, bem como de aprimorar nossas habilidades em análise de dados e programação no contexto do software MATLAB.

É relevante mencionar que opto por utilizar a mesma sequência de DNA da tarefa anterior, que corresponde à enzima **Hexokinase**. Essa escolha não apenas facilita a comparação direta com os resultados anteriores, mas também garante que as conclusões não sejam afetadas por diferenças nas sequências, evitando qualquer extrapolação indevida. Além disso, o formato FASTA da sequência em análise é altamente adequado para os procedimentos propostos nesta tarefa, simplificando a manipulação e análise subsequente dos dados.

2 Implementação das funções

2.1 Etapa 1:Análise de uma Cadeia de Markov com Dados de Sequências de DNA

1. **Leitura do Arquivo de Sequência de DNA:**

   Na primeira etapa da tarefa, inicio a análise pela leitura do arquivo de sequência de DNA no formato FASTA. Para esse propósito, fiz uso da função `fastaread`, uma escolha criteriosa em virtude de sua capacidade de separar adequadamente o cabeçalho do arquivo do sequenciamento propriamente dito. Essa distinção entre o cabeçalho e a sequência é crucial para a análise subsequente, pois permite isolar com precisão as informações relevantes do DNA a serem processadas. Essa abordagem difere da alternativa que utiliza o comando `fscan`, o

qual, ao contrário da função `fastaread`, mesclaria os dois segmentos do arquivo, potencialmente comprometendo, ainda que minimamente, o cálculo da entropia da sequência de DNA devido à possível interferência do cabeçalho no processamento dos dados.

2. **Definição dos Estados:**

   Para representar os estados da cadeia de Markov, opto por uma abordagem simples e direta, considerando os quatro principais nucleotídeos presentes no DNA como estados fundamentais: adenina (A), citosina (C), guanina (G) e timina (T). Essa escolha se pautou na simplicidade conceitual e na relevância biológica desses elementos nucleotídicos. Esses estados nucleotídicos foram, portanto, adotados como a base essencial para a análise subsequente, na qual explorei as transições entre eles para compreender melhor os padrões e comportamentos subjacentes à sequência de DNA.

3. **Contagem das Transições:**

   Com os estados definidos, prossegui para a contagem das transições entre esses estados. Para realizar essa tarefa, prossegui inicializando uma matriz denominada `transition_counts`. Essa matriz foi dimensionada para ter um tamanho de `num_states` por `num_states`, onde `num_states` representa o número de estados, ou seja, no nosso caso, os quatro nucleotídeos (A, C, G e T). Então, ela foi criada com valores iniciais zero e serviu para registrar as contagens das transições entre os estados.Para realizar a contagem das transições, iterei através da sequência de DNA em um loop `for`. A variável `i` representa a posição atual na sequência, variando de 1 até o comprimento da sequência menos 1, pois estamos interessados nas transições entre os nucleotídeos consecutivos. Dentro do loop, capturei o nucleotídeo atual (`current_nucleotide`) e o próximo nucleotídeo (`next_nucleotide`) na sequência. Em seguida, mapeei esses nucleotídeos para seus índices correspondentes no vetor `states` utilizando a função `find` e `strcmp`. Isso me permitiu converter os nucleotídeos em números que representam os estados. Por fim, atualizei a contagem de transição na matriz `transition_counts`. A cada transição encontrada, incrementei o valor na posição apropriada da matriz, ou seja, atualizei a contagem de transição do estado atual para o próximo estado. Essa abordagem no MATLAB permitiu uma contagem eficiente e precisa das transições entre os estados nucleotídicos, fornecendo uma base sólida para as análises subsequentes.

4. **Cálculo das Probabilidades de Transição:**

   Para obter uma visão mais probabilística das transições entre os estados nucleotídicos na cadeia de Markov, realizei o cálculo das probabilidades de transição. Essas probabilidades representam a chance de transição de um estado para outro, cujo cálculo foi realizado de forma eficiente no MATLAB. Inicialmente, dividi as contagens de transição na matriz `transition_counts` pela soma das contagens em cada linha, e o resultado desse cálculo foi armazenado na matriz `transition_probabilities`, que passou a conter as probabilidades de transição entre os estados. Para uma melhor visualização e interpretação dessas probabilidades, criei uma tabela chamada `transition_table`. Essa tabela foi configurada de forma que os rótulos das linhas e colunas fossem os estados nucleotídicos (A, C, G e T), tornando a representação das probabilidades de transição mais clara e organizada. Por fim, para apresentar a matriz de transição resultante de uma maneira fácil de ler, utilizei a função `disp` para exibir a tabela `transition_table`. O resultado foi uma representação tabular da matriz de transição da cadeia de Markov, permitindo uma inspeção visual das probabilidades de transição entre os estados nucleotídicos.

5. **Geração de Sequência de Estados:**

   Para compreender como a cadeia de Markov se comporta na prática, realizei a geração de uma sequência de estados utilizando a função `generateMarkovSequence`. Essa sequência foi criada com base na matriz de transição que representa as probabilidades de transição entre os estados nucleotídicos, obtida no passo anterior. Primeiramente, escolhi um estado inicial para iniciar a sequência. Neste caso, defini o estado inicial como 1, mas poderia ter escolhido qualquer estado (1, 2, 3 ou 4) com base na interpretação biológica ou em requisitos específicos da análise. Em seguida, determinei o comprimento da sequência desejada, que foi definido como 1000 estados no roteiro da tarefa em questão. A sequência foi gerada pela função `generateMarkovSequence`, que aceitou como entrada a matriz de transição (`transition_table`), o estado inicial (`initialState`) e o comprimento da sequência. A função `generateMarkovSequence` inicia convertendo a tabela de transição (`transitionTable`) em uma matriz numérica (`transitionMatrix`). Essa etapa é importante para facilitar os

cálculos subsequentes. Em seguida, a função verifica o número de estados na matriz de transição (`numStates`), garantindo a flexibilidade para trabalhar com cadeias de Markov de diferentes tamanhos. A partir de então, a sequência de estados é inicializada como um vetor de zeros com o comprimento especificado (`sequenceLength`), criando um espaço para armazenar os estados gerados. O estado inicial é definido com base no valor fornecido (`initialState`), que representa o primeiro estado da sequência. O coração da função está no loop `for`, que gera a sequência de estados. No interior do loop, o seguinte é realizado: São calculadas as probabilidades de transição a partir do estado atual. Isso é feito consultando a linha correspondente à `currentState` na matriz de transição (`transitionMatrix`). O próximo estado é escolhido com base nas probabilidades de transição. Isso ocorre de forma estocástica, onde cada estado tem uma probabilidade de ser escolhido proporcional à probabilidade de transição. O estado atual é atualizado para ser o próximo estado escolhido, e então é armazenado na sequência, atualizando a posição `i` da sequência com o novo estado gerado. O loop continua até que a sequência atinja o comprimento desejado, conforme especificado por `sequenceLength`. Ao final, a função retorna a sequência de estados gerada. Então, em posse desses estados, faço o mapeamento dos números de volta para os rótulos de estado e exibo a sequência em formato de texto para facilitar a visualização e o entendimento do resultado da função.

6. **Cálculo da Distribuição Estacionária:**

   Dando prosseguimento ao roteiro, foi necessário calcular a distribuição estacionária da cadeia de Markov, que representa o equilíbrio a longo prazo da cadeia. Esse cálculo foi realizado em etapas: Primeiramente, utilizei a função `eig` para realizar a diagonalização da matriz de transição (`transition_probabilities`). Essa função é usada para encontrar os autovalores e autovetores correspondentes da matriz. Os autovalores fornecem informações cruciais sobre o comportamento da cadeia de Markov. Após essa diagonalização, os autovalores foram armazenados em `D`, e os autovetores correspondentes foram armazenados em `V`. Esses autovetores representam as distribuições de probabilidade em estados estacionários. Para identificar a distribuição estacionária, localizei o índice do autovalor mais próximo de 1 (indicativo do estado estacionário) calculando a diferença entre os autovalores e 1 e encontrando o índice com a menor diferença absoluta. Isso foi realizado com as linhas de código `[ , index] = max(abs(diag(D)) - 1)`. Com o índice do autovetor correspondente à distribuição estacionária identificado, extraí a distribuição estacionária em si. Ela foi obtida tomando os valores absolutos dos elementos da coluna correspondente na matriz `V` e normalizando essa distribuição pela soma de seus valores, garantindo que a soma das probabilidades seja igual a 1. Para apresentar a distribuição estacionária de forma clara, criei uma tabela chamada `DistribuicaoEstacionaria` usando a função `array2table`. Nessa tabela, utilizei os nomes dos estados (A, C, G e T) como rótulos de linha e atribuí o nome `'Distribuicao_Estacionaria'` à variável de coluna. Por fim, exibi a distribuição estacionária como uma tabela na tela para permitir uma análise fácil e uma compreensão clara dos valores de probabilidade associados a cada estado.

7. **Verificação de Consistência da Distribuição Estacionária:**

   Após calcular a distribuição estacionária a partir da matriz de transição, é fundamental verificar sua consistência para garantir a precisão dos resultados. A consistência é importante porque a distribuição estacionária idealmente deve satisfazer a relação de que é o vetor próprio correspondente ao autovalor 1 da matriz de transição. Para realizar essa verificação, executei as seguintes etapas: Primeiramente, calculei a distribuição estacionária teórica multiplicando a distribuição estacionária original (`stationary_distribution`) pela matriz de transição (`transition_probabilities`). Em seguida, utilizei a função `ismembertol` para verificar a consistência entre a distribuição estacionária teórica calculada e a distribuição estacionária original. Essa função verifica se os dois conjuntos são iguais com uma tolerância especificada. Neste caso, a tolerância foi definida como `1e-6`, o que significa que as duas distribuições são consideradas iguais se a diferença entre seus elementos for menor que esse valor. O resultado da verificação é armazenado na variável `is_consistent`, que será um vetor lógico com o mesmo número de elementos que as distribuições. Se todos os elementos desse vetor forem `true`, significa que a distribuição estacionária é consistente com a matriz de transição. A seguir, exibi o resultado da verificação na tela. Se `is_consistent` for `true` para todos os elementos, uma mensagem é exibida indicando que a distribuição estacionária é consistente com a matriz de transição. Além disso, uma explicação é fornecida para esclarecer o motivo da consistência. No entanto, se `is_consistent` contiver pelo menos um `false`, é exibida

uma mensagem indicando que a distribuição estacionária não é consistente com a matriz de transição. A vantagem dessa verificação de consistência é garantir a validade dos resultados obtidos. A distribuição estacionária ideal representa o estado de equilíbrio da cadeia de Markov, e sua consistência com a matriz de transição é um indicativo de que os cálculos foram feitos corretamente. Qualquer inconsistência pode indicar erros nos cálculos ou na implementação. Além disso, fornecer essa verificação e explicação na saída do programa é útil para documentar e comunicar de forma transparente a qualidade dos resultados aos usuários ou leitores do código.

2.2 Etapa 2: Cálculo da Taxa de Entropia

1. **Implementação da Função de Entropia e Cálculo da entropia da Matriz de Transição:**

   Para calcular a entropia da fonte de informação, criei uma função chamada `calculateEntropy`. Esta função foi projetada para ser versátil, permitindo o cálculo da entropia tanto para sequências de caracteres quanto para matrizes numéricas, conforme solicitado na tarefa. Primeiramente, a função verifica o tipo de entrada usando as estruturas condicionais `if` e `elseif`. Ela aceita dois tipos de entrada: Se a entrada for uma sequência de caracteres (`ischar(input_data)`), ela converte a sequência de caracteres para maiúsculas usando `upper(input_data)`, garantindo que caracteres maiúsculos e minúsculos sejam tratados da mesma forma, e remove caracteres não-alfabéticos usando a expressão regular `regexprep(...)`. Isso elimina qualquer caractere que não seja uma letra do alfabeto. Em seguida, ela calcula as frequências dos caracteres no texto usando `histc` e converte essas frequências em probabilidades `p`. Calcula, então, a entropia usando a fórmula da entropia de Shannon:

   $$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)) \tag{1}$$

   Por outro lado, se a entrada for um número (`isnumeric(input_data)` ou uma matriz numérica (`ismatrix(input_data)`), a função calcula a entropia diretamente, sem a necessidade de pré-processamento. Por fim, se a entrada não for nem uma sequência de caracteres nem um número ou uma matriz numérica, a função gera um erro indicando que o tipo de entrada não é suportado.

   Para cumprir o que foi solicitado no roteiro da tarefa, apliquei a matriz de transição à função `calculateEntropy` para calcular sua entropia. Cito que essa matriz contém as probabilidades de transição entre os estados da cadeia de Markov, de modo que cada elemento representa a probabilidade de transição de um estado para outro. O resultado deste cálculo é armazenado na variável `entropy_transition`, que contém, portanto, a taxa de entropia da matriz de transição, dada por `entropy_transition`=**7,8419**. Em seguida, utilizamos a função `disp` para exibir a taxa de entropia calculada na tela. A mensagem "Taxa de Entropia da Matriz de Transição:"é exibida para identificar o valor que está sendo apresentado. Este cálculo de entropia nos fornece uma medida da incerteza ou desordem presente nas transições entre os estados da cadeia de Markov, o que é fundamental para a análise da sequência de DNA em questão.

2. **Cálculo da Taxa de Entropia da Cadeia de Markov na Parte 1:**

   Além da matriz de transição, também calculei a taxa de entropia para a Cadeia de Markov representada pela sequência de estados gerada na Etapa 1 do projeto.

   Primeiramente, utilizamos a função `calculateEntropy` para calcular a entropia. Essa função, como foi explicitado, é versátil e pode ser aplicada tanto a sequências de caracteres quanto a matrizes numéricas. No nosso caso, a entrada fornecida à função foi a sequência de estados representada por `sequence_text`.

   Para entender melhor o cálculo da entropia, é importante mencionar que a entropia de Shannon é uma medida da incerteza ou desordem presente em um sistema. No contexto da Cadeia de Markov, ela nos indica o nível de incerteza nas transições entre os estados da cadeia.

   Após o cálculo, o valor da entropia da sequência gerada da Cadeia de Markov é armazenado na variável `entropy_Markov`. Neste caso, o valor obtido foi de **1,9693**, indicando o nível de incerteza na sequência de estados.

Em seguida, utilizo a função `disp` para exibir a taxa de entropia calculada na tela. A mensagem "Cadeia de Markov em análise"é exibida para identificar a sequência de estados que está sendo analisada (dada a natureza estocática da saída em questão). A mensagem "Taxa de Entropia"é exibida seguida do valor da entropia, fornecendo assim uma medida quantitativa da incerteza presente na Cadeia de Markov.

2.3 Comparação com a Propriedade da Equipartição Assintótica

Os resultados obtidos na análise da Cadeia de Markov aplicada à sequência de DNA fornecem insights significativos sobre a distribuição de estados nesse contexto. No entanto, é crucial contrastar esses resultados com as conclusões anteriores sobre a Propriedade da Equipartição Assintótica.

No trabalho anterior, observou-se que, para sequências curtas, o modelo de grande número de eventos raros não se aplica adequadamente. Em vez disso, identificou-se a Lei de Zipf, na qual algumas características (ou sequências de nucleotídeos) são consideravelmente mais frequentes do que outras. Isso sugere que certos elementos ocorrem com alta frequência, enquanto a maioria dos elementos é rara.

Por outro lado, à medida que a sequência genômica se torna mais extensa, a Propriedade da Equipartição Assintótica começa a prevalecer. Ela postula que, em conjuntos de dados geneticamente grandes o suficiente, todas as sequências de nucleotídeos tendem a ter frequências aproximadamente iguais. Isso significa que, conforme a sequência se estende, a distribuição de frequência se aproxima da equipartição, na qual todas as sequências têm uma frequência semelhante.

Ao comparar essas conclusões com os resultados da análise da Cadeia de Markov, podemos observar como a distribuição de estados na sequência de DNA se comporta em diferentes cenários de comprimento. A taxa de entropia calculada para a Cadeia de Markov oferece uma medida da incerteza ou desordem nas transições entre os estados, permitindo uma compreensão mais aprofundada de como a informação está distribuída ao longo da sequência de DNA.

Essa análise comparativa enriquece nossa compreensão da organização da informação genômica e pode ser fundamental para a interpretação de resultados em estudos genéticos e biológicos.

3 Apresentação dos resultados obtidos

Antes de iniciar, de fato, a análise, cito que os valores propriamente ditos das probabilidas, as sequências geradas e as entropias calculadas, por terem uma natureza estocástica, apresentarão diferenças caso a caso. No entanto, a proposta da utilização do script será conservada.

A análise da Cadeia de Markov aplicada à sequência de DNA revelou informações importantes sobre a distribuição de estados nesse contexto. A matriz de transição da Cadeia de Markov para o presente caso está representada abaixo:

	A	C	G	T
A	0.38049	0.18913	0.2137	0.21668
C	0.3465	0.2088	0.24492	0.19977
G	0.31514	0.28836	0.19773	0.19876
T	0.23241	0.1791	0.29318	0.29531

Além disso, temos a sequência de estados gerada em formato de texto:
`AGCGTCATCGAA...AACATCAAACACTTAGCGCACTTAAATGGAATGCCCAATTGCGGTACCGCAAAAAGCGTGC`

Nota-se que a distribuição estacionária calculada é consistente com a matriz de transição, o que indica que a análise foi realizada corretamente. A distribuição estacionária é dada por:

Estado	Distribuição Estacionária
A	0.3243
C	0.21436
G	0.23466
T	0.22667

Tal verificação de consistência é confirmada porque a distribuição estacionária é dada pelo vetor próprio correspondente ao autovalor 1 da matriz de transição, e os resultados coincidem:

Distribuição Estacionária	Distribuição Calculada da Matriz de Transição
0.3243	0.3243
0.2144	0.2144
0.2347	0.2347
0.2267	0.2267

Na segunda parte da análise, é calculada a taxa de entropia da matriz de transição, que foi de 7.8419. Isso nos fornece uma medida da incerteza ou desordem presente nas transições entre os estados da Cadeia de Markov, sendo notoriamente um valor bastante elevado.

Além disso, foi calculada a taxa de entropia para a sequência de estados gerada, que foi de 1.9679. Comparando esses resultados com as conclusões anteriores sobre a Propriedade da Equipartição Assintótica, podemos observar como a distribuição de estados na sequência de DNA se comporta em diferentes cenários de comprimento.

Os resultados indicam que a matriz de transição possui uma entropia significativamente maior do que a sequência de estados gerada, o que sugere que a matriz de transição apresenta uma maior incerteza nas transições entre os estados. Isso pode estar relacionado ao fato de que a matriz de transição é uma representação mais abstrata da sequência de DNA e não leva em consideração os detalhes locais da sequência.

4 Discussão sobre as implicações dos resultados obtidos em termos da quantidade de informação presente na sequência de DNA e sua relevância no contexto genômico.

Com o decorrer do presente trabalho prático, conclui-se que os resultados da análise da Cadeia de Markov aplicada à sequência de DNA revelam informações valiosas sobre a quantidade de informação presente nesse material genético e suas implicações no contexto genômico.

Primeiramente, ao examinar a matriz de transição da Cadeia de Markov, pode-se observar como as bases A (Adenina), C (Citosina), G (Guanina) e T (Timina) estão relacionadas nas transições entre estados. A matriz de transição revela as probabilidades de transição entre essas bases, indicando quais são mais propensas a seguir umas às outras. Por exemplo, a probabilidade de transição de A para G é de 0.2137, enquanto a de C para G é de 0.24492. Isso nos mostra como as bases são organizadas na sequência de DNA e como certas combinações são mais frequentes do que outras.

Essas informações têm implicações significativas no contexto genômico. Por exemplo, uma compreensão detalhada das probabilidades de transição entre bases pode ser fundamental na identificação de regiões codificadoras de genes. Se determinadas combinações de bases ocorrem com maior frequência em regiões codificadoras, isso pode indicar a presença de genes importantes. Essa é uma das maneiras pelas quais a análise de Cadeias de Markov pode auxiliar na anotação genômica.

Além disso, a distribuição estacionária da Cadeia de Markov fornece informações sobre a frequência relativa de cada base na sequência de DNA. Por exemplo, a distribuição estacionária indica que a base A ocorre com uma frequência de 0.3243, enquanto a base C ocorre com 0.21436. Essa distribuição de frequência pode ser comparada com a frequência observada em um conjunto de dados genômicos real. Se a distribuição estacionária for consistente com a frequência observada, isso sugere que a sequência de DNA está em equilíbrio, o que é um aspecto importante na análise genômica.

No entanto, a taxa de entropia da matriz de transição e da sequência de estados gerada fornece informações adicionais. A matriz de transição possui uma entropia significativamente maior do que a sequência de estados gerada. Isso indica que a matriz de transição apresenta uma maior incerteza nas transições entre os estados. Por outro lado, a sequência de estados gerada tem uma entropia menor, o que sugere que a sequência possui um grau de previsibilidade maior em comparação com a matriz de transição.

Essa diferença na entropia pode ser crucial na análise de sequências de DNA. Por exemplo, regiões genômicas altamente conservadas, como exons de genes, tendem a ter uma entropia menor, uma vez que as bases são altamente previsíveis. Em contraste, regiões não codificadoras ou regiões regulatórias podem apresentar uma entropia maior devido à presença de elementos regulatórios variáveis.

Em resumo, a análise da Cadeia de Markov oferece uma visão profunda da organização e distribuição das bases em sequências de DNA. Essas informações têm implicações significativas na

anotação genômica, na identificação de regiões codificadoras e não codificadoras, e na compreensão da variabilidade genômica. A combinação de dados sobre frequência, entropia e transições entre estados fornece uma perspectiva abrangente sobre como a informação genética está distribuída e organizada em sequências de DNA.

Referências

1. https://doi.org/10.1002/9781118033296.ch17
2. https://en.wikipedia.org/wiki/Hexokinase
3. https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/hexokinase
4. https://www.sciencedirect.com/topics/earth-and-planetary-sciences/hexokinase
5. https://proteopedia.org/wiki/index.php/Hexokinase
6. https://www.ebi.ac.uk/interpro/entry/InterPro/IPR022672/
7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4291497/