Elementos de Programação

Projecto de Biocomputação

Departamento de Matemática, IST

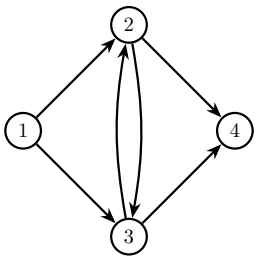
Novembro de 2020

A experiência de Adleman

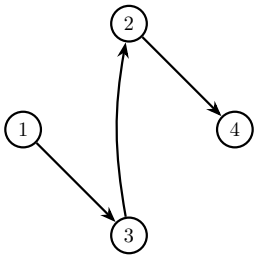
Em 1994, Leonard Adleman surpreendeu a comunidade científica quando publicou (Molecular computation of solutions to combinatorial problems, *Science* 226, 1021-1024, 1994) os resultados da experiência que veio a ficar conhecida com o seu nome.

Adleman resolveu com técnicas de biologia molecular o **problema HPP** (*Hamiltonian Path Problem*) a seguir descrito. Dado um grafo dirigido e fixado um vértice origem e um vértice destino, pretende encontrar-se, caso exista, um caminho Hamiltoniano com essa origem e esse destino. Por caminho Hamiltoniano entende-se um caminho que passa por todos os vértices uma e uma só vez.

Para ilustração, considere-se o grafo:



Neste grafo, existem dois caminhos Hamiltonianos. Por exemplo,



Este problema é difícil de resolver com meios computacionais convencionais, pois envolve a pesquisa exaustiva de todas as possíveis soluções. A proposta de Adleman permite tirar partido de um elevado nível de paralelismo para chegar à solução (se existir) mais rapidamente. A ideia geral é simples - representam-se os vértices e as arestas por sequências de nucleótidos de tal maneira que a hibridação leve à concatenação de arestas contíguas no grafo.

Recorde-se que o DNA é constituído por: A (*adenina*), C (*citocina*), T (*timina*) e G (*guanina*). Recorde-se também que A e T são complementares no sentido em que na hibridação eles se emparelham através de duas ligações de hidrogénio e que C e G são complementares com emparelhamento através de três ligações de hidrogénio.

Adleman codificou cada vértice por uma sequência de nucleótidos de comprimento par. Por exemplo, sejam os vértices u, v e w representados pelas sequências U_1U_2, V_1V_2 e W_1W_2, respectivamente, em que U_1, U_2, V_1, V_2, W_1 e W_2 são sequências de comprimento dez. Então, as arestas $u \to v$ e $v \to w$ deverão ser representadas pelas sequências $\overline{U_2V_1}$ e $\overline{V_2W_1}$, repectivamente. Aqui, dada uma sequência Z de nucleótidos, $\overline{Z}$ é a sequência dos seus complementares. Por exemplo, se $Z = \text{ACATGG}$, então $\overline{Z} = \text{TGTACC}$.

Numa *sopa biomolecular* em que estejam presentes as sequências V_1V_2, $\overline{U_2V_1}$ e $\overline{V_2W_1}$, por hibridação, acabará por se construir a trança aberta

$$\frac{V_1\,V_2}{\overline{U_2}\,\overline{V_1}\,\overline{V_2}\,\overline{W_1}}\ .$$

Ou seja, surgirá o caminho $u \to v \to w$ por concatenação dos caminhos $u \to v$ e $v \to w$.

Dados um grafo dirigido e fixado o vértice partida e o vértice de chegada, uma vez estabelecida esta codificação dos vértices e das arestas, a experiência de Adleman procede do modo seguinte:

Passo 0: Preparação
Constitui-se uma solução biomolecular num tubo de ensaio com grande quantidade de sequências de nucleótidos representando os vértices e as arestas do grafo dirigido. Adleman usou 10^{13} cópias de cada vértice e de cada aresta.

Passo 1: Hibridação
Deixa-se o mecanismo de hibridação actuar, o que conduz à constituição de muitas tranças representando caminhos possíveis no grafo dirigido em questão. É neste passo que o método de Adleman introduz elevados níveis de paralelismo pois as tranças vão-se constituindo ao mesmo tempo.

Passo 2: Selecção dos caminhos com origem e destino pretendidos
Recorrendo à técnica *PCR* (de *Polymerase Chain Reaction*), eliminam-se as tranças que não têm a origem e o destino pretendidos.

Passo 3: Selecção dos caminhos com o comprimento pretendido
Recorrendo à técnica *GE* (de *Gel Electrophoresis*), eliminam-se as tranças que não têm comprimento correspondente ao número de vértices do grafo dirigido em causa.

Passo 4: Selecção dos caminhos que passam em todos os vértices
Para cada vértice, recorrendo a uma técnica de *purificação por afinidade*, eliminam-se as tranças que não passam por esse vértice.

Passo 5: Observação do resultado
Procede-se à sequenciação das tranças de DNA que tenham sobrevivido aos passos anteriores. Qualquer delas representa um caminho Hamiltoniano no grafo dirigido dado com a origem e destino pretendidos.

Para mais detalhes sobre a experiência de Adleman, recomenda-se a visita à página
`https://archive.arstechnica.com/reviews/2q00/dna/dna-1.html`.

Simulação da experiência de Adleman

O objectivo do projecto é desenvolver em `Python` um programa que o resolva o *HPP* (*Hamiltonian Path Problem* - Problema do caminho Hamiltoniano) através de uma simulação abstracta da experiência de Adleman, de acordo com os princípios de simulação discreta estocástica. Para o efeito, considera-se que a solução de sequências de nucleótidos se encontra num cubo de dimensões $[0,1]^3$, em que cada sequência corresponde a um caminho no grafo. Esta solução evolui de acordo com os passos descritos anteriormente, que a seguir se detalham, e em que a hibridação é representada por simples concatenação de caminhos, sem entrar nos detalhes da codificação em sequências de nucleótidos.

Passo 0: Preparação
Distribui-se de modo aleatório uniforme κ cópias de cada uma das arestas do grafo dirigido G dado no cubo $[0,1]^3$. Cada uma destas cópias corresponde a um indivíduo (um caminho numa determinada posição do cubo) que irá evoluir durante o processo de hibridação como a seguir se descreve.

Passo 1: Hibridação
Neste passo, o sistema evolui até um tempo limite de simulação *ht* por ocorrência de eventos locais que podem ser de concatenação ou deslocamento, ou eventos globais de cisão:

- **Concatenação**, com tempo médio entre concatenações *tbc*. Um indivíduo combina-se com outro indivíduo escolhido aleatoriamente de entre os 5 indivíduos compatíveis mais próximos. Dois indivíduos dizem-se compatíveis se o último vértice do caminho de um deles é igual ao primeiro vértice do caminho do outro. Da combinação destes dois indivíduos resulta um novo indivíduo com caminho correspondente à junção dos dois caminhos anteriores e colocado na posição média dos indivíduos originais, que devem ser retirados da solução.

- **Deslocamento**, com tempo médio entre deslocamentos *tbd*. Um indivíduo na posição (x, y, z) e com um caminho de comprimento c desloca-se para uma nova posição no cubo escolhida aleatoriamente em $[x-d, x+d] \times [y-d, y+d] \times [z-d, z+d]$, onde $d = 1/c$, garantindo que a nova posição se encontra no cubo.

- **Cisão**, com tempo médio entre cisões *tbz*. Este evento global percorre os indivíduos na solução e aqueles cujo comprimento do caminho exceda o comprimento desejado (o número de vértices) são sucessivamente separados em novos indivíduos colocados aleatoriamente no cubo. O caminho de cada um destes novos indivíduos corresponde a um fragmento do caminho original de comprimento aleatório que não deverá exceder metade do comprimento desejado.

Passo 2: Selecção dos caminhos com origem e destino pretendidos
Eliminam-se os caminhos que não têm a origem o e o destino d pretendidos.

Passo 3: Selecção dos caminhos com o comprimento pretendido
Eliminam-se os caminhos que não têm comprimento igual ao número de vértices de G.

Passo 4: Selecção dos caminhos que passam em todos os vértices
Para cada vértice de G, eliminam-se os caminhos que não passam por esse vértice.

Passo 5: Observação do resultado
Apresentam-se os caminhos que tenham sobrevivido aos passos anteriores. Qualquer deles é um caminho Hamiltoniano no grafo dirigido G com origem o e destino d.

Desenvolva o programa de simulação seguindo o *método de programação modular por camadas centradas nos dados*:

1. Comece por identificar os objectos de trabalho, nomeadamente grafos, caminhos, indivíduos, eventos, cadeia de acontecimentos pendentes e solução biomolecular.
2. Implemente esta camada sobre a camada básica da linguagem `Python`.
3. Desenvolva de seguida o programa abstracto pretendido sobre a camada que disponibiliza estes objectos.
4. Integre o programa obtido em 3 com os módulos desenvolvidos em 2 para obter o programa final.
5. Experimente o programa com diversos conjuntos de dados.

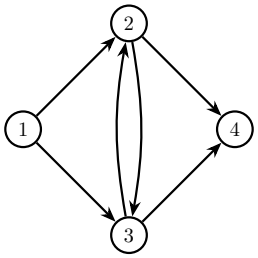
Dados fornecidos

O programa deve receber os dados seguintes:

- G – grafo dirigido;
- o – vértice origem do caminho Hamiltoniano pretendido;
- d – vértice destino do caminho Hamiltoniano pretendido;
- κ – número de cópias a utilizar (muito inferior ao número adoptado por Adleman - 10^{13});
- ht – tempo limite de simulação;
- tbc, tbd, tbz – tempo médio entre concatenações, tempo entre deslocamentos e tempo entre cisões, respectivamente.

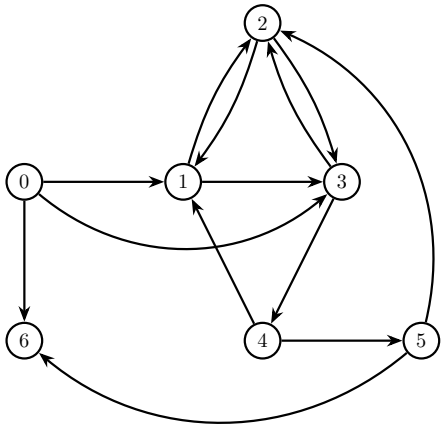
Começe por experimentar o programa com os seguintes conjuntos de dados:

Conjunto 1 (grafo considerado acima)



- $o = 1$;
- $d = 4$;
- $\kappa = 20$;
- $ht = 20$;
- $tbc = tbd = tbz = 1$.

Conjunto 2 (grafo considerado por Adleman)



- $o = 0$;
- $d = 6$;
- $\kappa = 150$;
- $ht = 100$;
- $tbc = 1$;
- $tbd = 1$;
- $tbz = 10$.

Resultados pretendidos

O programa deve apresentar o conjunto dos caminhos Hamiltonianos encontrados.

Devem ser apresentados os resultados obtidos com os dois conjuntos de dados obrigatórios e ainda com, pelo menos, um terceiro conjunto de dados de escolha livre.

Entrega do projecto

O projecto é entregue através do sistema Fenix, após a inscrição do respectivo grupo. A entrega do projecto está dividida em duas partes.

Parte 1

Na primeira parte do projecto, cada grupo deve submeter os módulos desenvolvidos no ponto 2 da secção anterior. A entrega deve consistir de um único arquivo (zip ou rar) contendo os módulos desenvolvidos, e um pequeno relatório descrevendo as operações dos tipos de dados e explicando as principais opções tomadas para a sua implementação, bem como exemplos ilustrando o seu correcto funcionamento.

Data limite de submissão: **23h59m do dia 12 de Dezembro de 2020**.

Parte 2

Na segunda parte do projecto, cada grupo deve submeter o simulador. Para esta fase, serão disponibilizados na página da disciplina módulos com implementações dos tipos de dados. Cada grupo pode optar por desenvolver o simulador recorrendo aos módulos disponibilizados ou recorrendo aos seus próprios módulos, possivelmente alterados após a primeira submissão. Tal opção deve estar claramente identificada no relatório, implicando a ressubmissão dos elementos da Parte 1, caso os tipos de dados tenham sido alterados e utilizados. A entrega deve consistir de um único arquivo (zip ou rar) contendo o simulador e eventuais módulos adicionais que tenham sido desenvolvidos, e um pequeno relatório explicando as principais opções tomadas para a implementação do simulador e exemplos que ilustrem e permitam analisar o comportamento do modelo proposto.

Data limite de submissão: **23h59m do dia 9 de Janeiro de 2021**.