Trabalho 3 de Bioinformática

Análise Automática de Sequências

20 de Maio de 2019

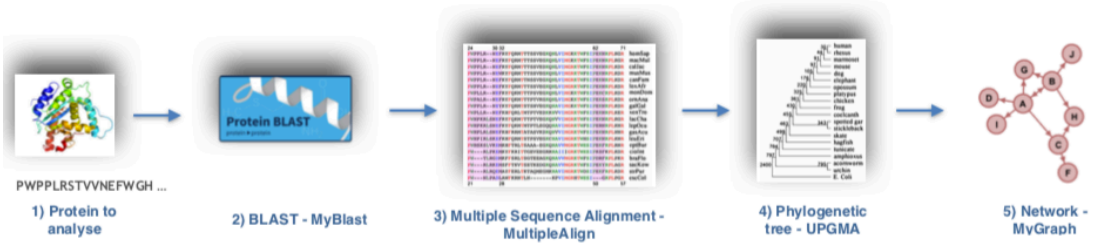
Entrega: 14/06/2019 até 23:59

Este trabalho é para ser submetido via Moodle. O trabalho será desenvolvido como trabalho extra-classe. Os testes e exames poderão conter perguntas relacionadas com este trabalho. Para saber o método e critério de avaliação, por favor consulte a ficha da unidade curricular na página do sigarra.

1 Descrição do Problema

Neste trabalho deverá consolidar os conhecimentos das aulas de pesquisa de sequências similares em bases de dados, alinhamento múltiplo de sequências, árvores filogenéticas e grafos e redes biológicas.

Para o efeito deverá utilizar, alterar e estender com novas funcionalidades o código fornecido e desenvolvido nas aulas. Neste trabalho é dado um maior grau de liberdade e compete ao aluno tomar as decisões mais apropriadas para a sua concretização. Decisões e novas ideias implementadas devem ser discutidas no relatório.



O objectivo do trabalho é implementar uma "*pipeline*" bioinformática que execute os passos descritos na figura acima. Uma "pipeline" refere-se a uma *script* que executa de forma automática e encadeada uma série de passos até chegar a uma resultado final.

1) Sequência a analisar
É fornecida no ficheiro *source.fasta* a sequência a analisar. Neste caso já conhecemos a função da sequência e a espécie da mesma é indicada entre [].

2) Análise Blast
É fornecido um conjunto de sequências no ficheiro *seqdump.txt* que irão formar a base de dados de sequências a analisar. O objectivo deste passo é encontrar as dez sequências mais similares à sequência *source*. Estas sequências deverão ser de espécies diferentes tal como indicado no campo entre [].

Notas: i) Poderão usar os valores de parâmetros tal como definidos no código das aulas ou alterar os mesmos se necessário. Indiquem e discutam as vossas escolhas.
ii) A função best_alignment retorna o melhor alinhamento encontrado na *db*. Poderá alterar esta função para ir encontrando o melhor alinhamento, remover a respetiva sequência e correr de novo o método até as dez sequências serem encontradas.

3) Alinhamento Múltiplo
Produza um alinhamento múltiplo do conjunto de 11 sequências (source + 10 target encontradas). Deverá visualizar o alinhamento. Para esta análise e subsequentes deverá utilizar a informação referente à espécie da sequência e substituir os espaços por *underscore.*

4) Árvore
Produza e visualize a árvore filogenéticas das 11 sequências (source + 10 target encontradas).

4) Grafo
Deverá desenvolver um método que gere um grafo com a relação das distâncias entre as sequências. Este método terá por base a matriz de distâncias (*pairwise distance*) geradas no passo anterior. Deverá receber como argumento esta matriz e um valor de corte de distâncias. Este valor de corte indica que pares de sequências com distâncias superiores a este valor não serão consideradas conectadas. Sendo apenas consideradas ligadas entre si se a distância for inferior a este valor. A partir daqui deverá gerir o respetivo grafo de ligação entre sequências, visualizar e obter todas as métricas do mesmo. Experimente diferentes valores de corte para obter uma rede minimamente interessante.
Nota: O grafo será não direcionado e não pesado. As distâncias servem apenas para definir o critério de ligação entre as sequências.

```
Input: matDist, cut_dist
     For each x and y in matDist:
          if dist(x,y) < cut_dist then x and y are connected
```

Deverá gerar visualizações e/ou listagens para todos os passos anteriores.

2 Relatório para entrega
O trabalho deverá ser acompanhado de um pequeno relatório (em Português ou Inglês) com o máximo de duas páginas (tamanho de letra 11) e em formato pdf. Neste deve discutir os seguintes pontos:

- Introdução - Contextualizar e descrever brevemente o problema.

- Descrição e estratégias de implementação - Discutir abordagens relevantes ao problema. Deverá incluir uma tabela com os valores e explicação de cada um dos parâmetros usados em cada etapa.

- Resultados - Deve indicar que funcionalidades foram implementadas, se conseguiu implementar todas as funcionalidades pedidas e se implementou outras funcionalidades além das especificadas. Poderá organizar esta informação usando um tabela.

- Comentários e Conclusões.

- Referências Bibliográficas (precisam ser explicitamente citadas no texto para saberem de onde o texto foi retirado/adaptado! Copiar é crime e poderá transformar-se em processo disciplinar, portanto evitem copiar textos e códigos. Se utilizarem figuras retiradas da web ou de livros ou de artigos etc, é necessário colocar uma referência explícita e clara. Por favor tenham atenção aos erros ortográficos.

Em anexo ao relatório deverá incluir prints das listagens e/ou visualizações geradas devidamente numeradas e legendadas. Deverá a partir do seu relatório referenciar estas mesmas imagens criadas.

Importante: deve gerar funções de demonstração completas e interessantes. Este é um dos elementos de avaliação mais importantes. Desta forma a *demo* do seu trabalho deverá evidenciar todo o trabalho desenvolvido e estar organizado de forma a ser perceptível a análise em questão.

3 Entrega

Submeter através do Moodle um arquivo zip contendo todo o código fonte dos programas e instruções de como executar('readme'). Todos os ficheiros devem ser colocados na mesma pasta incluindo os ficheiros com sequências de teste.

Importante: Deverão implementar um ficheiro run_me.py em que fazem a importação funções desenvolvidas e através de várias exemplos demonstram a chamada dos vários métodos implementados. Para tal o programa deve imprimir mensagens a indicar a funcionalidade implementada. **O programa deve correr na linha de comando (python run_me.py)**.

O trabalho pode ser feito em grupo de no máximo duas pessoas. Trabalhos com cópia de código de outros grupos serão desclassificados!

sequências de genes

source.fasta - sequência da proteína a analisar.
seqdump.txt - base de dados de sequências a comparar.