Implementação de algoritmos local e global

Bioinformática

António Sousa - up201208681

Bruno Cabral - up201202369

Ricardo Santos - up201203540

3 de Abril de 2018

1 Introdução

O alinhamento de pares de sequências nucleotídicas (i.e., genes) ou aminoacídicas (i.e., proteínas) permite determinar a similaridade entre duas sequências. Este processo é fundamental para determinar a função e estrutura de genes/proteínas com base na percentagem de similaridade entre duas sequências emparelhadas. Se o alinhamento entre duas sequências nucleotídicas/aminoacídicas é parcialmente ou completamente contíguo ao longo da sua extensão (i.e., poucas ou nenhumas "gaps"), então é muito provável que estas sequências representem genes/proteínas homólogas, i.e., descendam do mesmo antepassado.

Os algoritmos que implementam o alinhamento de pares de sequências nucleotídicas/aminoacídicas através de programação dinâmica começam por decompor as sequências nas suas unidades básicas, i.e., nucleotídeos/amino ácidos, para obter o melhor alinhamento entre as subunidades que compõem as duas sequências e, através desta forma, conseguir reconstruir o melhor alinhamento entre o par de sequências. Este problema exige a construção de uma matriz de similaridade, com um esquema de pontos ou "scores" (pontos diferentes para "match", "mismatch", "gap") associado. A penalização para a extensão de uma "gap" pode ser linear ou não. Portanto, o "score" do emparelhamento entre duas sequências é o somatório de "matches" e a subtração de "mismatches" e "gaps".

Os alinhamentos entre pares de sequências filogeneticamente próximas usualmente resultam em alinhamentos completos óptimos. Neste caso, o alinhamento global desenvolvido por Needleman and Wunsch (1970) é o melhor a aplicar. No entanto, o alinhamento entre pares de sequências muito divergentes resulta em maus alinhamentos globais devido às "gaps" nas margens das sequências que não emparelham. Para ultrapassar este problema, usualmente aplica-se o alinhamento local desenvolvido por Smith and Waterman (1981) que procura maximizar o melhor alinhamento entre subsequências, sem penalizar as "gaps" nas margens das sequências.

1.1 Objetivos do trabalho

Os objetivos deste trabalho consistiram na: (1) implementação do alinhamento global e local através de programação dinâmica; (2) utilização de um serviço web - REST - para testar remotamente os algoritmos desenvolvidos, i.e., alinhamento global e local, em bases de dados de genes e proteínas, como o European Nucleotide Archived (ENA) e o UniProt.

2 Implementação

Todo o trabalho foi desenvolvido em linguagem **Java**. No decorrer da implementação foram criados cinco ficheiros, cada uma com uma função específica.

- **Main.java**
- **Global.java**
- **Local.java**
- **Matriz.java**
- **Client_Pesquisa.java**
- **Client_Comparar.java**

Todas as classes são controladas pela classe **Main.java**, reponsável por lançar todos os métodos e pela interface.

```
Global(String s1, String s2)
Local(String s1, String s2)
```

Como o próprio nome indica a classe `Global` e `Local` foram implementadas para levar a cabo o alinhamento global e local. Dentro dos respetivos ficheiros existem métodos que permitem inicializar a matriz, calcular os "scores", fazer o alinhamento desejado e ainda

imprimir a matriz. No cálculo dos "scores", em caso de "match" soma-se 5 caso contrário subtrai-se o mesmo valor.

```
Matriz(String d, int v)
```

Esta classe guarda cada célula da matriz com uma "string" que vai identificar a direção que deu origem à respetiva célula bem com o seu valor.

```
procura(String tipo)
```

Dentro da classe **Client.java** é feito o pedido "online", quer este pedido se trate de uma proteína ou gene.
Compilar Programa:

```
javac Main.java
```

Correr:

```
java Main
```

Existem 3 tipos de interação possível:

1. **Introduzir sequências manualmente**

   Podemos simplesmente inserir as sequências pretendidas e de seguida escolher o tipo de alinhamento (local ou global).

2. **Pesquisa "online"**

   Permite escolher uma proteína ou gene através de um identificador específico, guardando o respectivo FASTA num ficheiro com o nome igual ao identificador. No caso da proteína, é feito um pedido a (ex. Q91502): `https://www.uniprot.org/uniprot/‘‘+param+’’.fasta`. Relativamente ao gene, o pedido é enviado para (ex. A00145): `https://www.ebi.ac.uk/ena/data/view/‘‘+param+’’&display=fasta` (em ambos os casos, o ***param*** é o identificador).

3. **Comparar sequências a partir de ficheiro**

   Para este caso, é necessário o cumprimento da Pesquisa Online que nos permite obter os ficheiros com o conteúdo pretendido.

   Para evitar este passo podem ser utilizados os ficheiros A00145.txt e A00146.txt que se encontram no mesmo diretório que os restantes ficheiros. Depois de os ficheiros terem sido carregados com sucesso é possível escolher entre executar um alinhamento offline ou online.

   O alinhamento offline utiliza a nossa implementação dos algoritmos global e local. Quando ao alinhamento online, é necessário especificar se a sequência se trata de um gene ou proteína, gerando um ficheiro output.txt com a informação obtida do serviço REST.

3 Funcionamento do Programa

Para correr o programa basta compilar e correr o programa usando o seguinte comando javac Main.java && java Main

Após a execução irá aparecer uma interface onde são demonstradas as funcionalidades do programa, são elas:

- A primeira opção, onde o utilizador insere duas sequências manualmente e usa os algoritmos criados, tanto o de alinhamento global como local, para obter um resultado;

- A segunda opção onde utilizando os serviços *Rest* é possível obter a sequência de uma proteína ou gene, se previamente o utilizador souber qual o nome, sendo que no fim é gerado um ficheiro Nome_Proteina_ou_Gene.txt onde é guardada toda a sequência

- Por último a última opção permite ao utilizador comparar dois ficheiros que contêm uma proteína ou gene, esta comparação pode ser feita de duas formas. Uma é utilizando os algoritmos que foram criados por nós para o alinhamento global e local e a outra faz uso do serviço *Rest* para que dado duas sequências, utilizando uma consulta a um servidor seja possível obter a resposta que é um conjunto de informação resultante desse alinhamento, essa informação é guardada num ficheiro gerado chamado output.txt.

4 Exemplo

Para este exemplo já foram efetuadas a Pesquisa Online pelos genes A00145 e A00146 sendo que os ficheiros .txt gerados foram feitos com recurso ao nosso programa. Basta compilar e correr o programa usando javac Main.java && java Main após isso pode ser selecionado o seguinte caminho

1. 3 (Enter)
2. A00145.txt (Enter)
3. A00146.txt (Enter)
4. 2 (Enter)
5. 2 (Enter)

após a consulta será gerado um output.txt com os resultados da comparação online que foi feita, usando o serviço Rest.

5 Conclusão

Com este trabalho foi possível implementar os algoritmos de alinhamento global e local de pares de sequências nucleotídicas/aminoacídicas, em linguagem Java, através

de programação dinâmica, com base nos trabalhos desenvolvidos por Needleman and Wunsch (1970) e Smith and Waterman (1981). A eficiência dos algoritmos desenvolvidos foi testada através do serviço remoto - REST - às bases de dados ENA e UniProt.

Referências

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195-197.