Francisca Conrado Silva
up201804188

Análise Automática de Sequências

Introdução

Compreender de que forma é que diversas sequências se relacionam umas com as outras, bem como poder representar essas relações em árvores filogenéticas e redes biológicas como grafos, só é possível com um conjunto de análises prévias.

Com o desenvolvimento da ferramenta de Blast, passou a ser possível perquisar num conjunto de sequências, como numa base de dados, aquelas que são mais semelhantes a uma dada sequência query de input. Ora, esta ferramente permite de forma rápida obter uma lista de sequências relacionadas entre si. Contudo, para ser possível a sua representação é necessário alinhar todas as sequências, tarefa que pode ser executada com o alinhamento global, dos investigadores Needleman e Wunch[1]. Seguidamente pode ser construída uma árvore filogenética e um grafo, sendo o último passo passar essas construções para uma representação esquemática e mais visual.

Este trabalho visa otimizar este processo, automatizando-o dando apenas uma sequência de referência e um ficheiro com diversas sequências, a base de dados. Para tal foram implementadas algumas alterações ao algoritmo original do alinhamento global, de forma a ser possível alinhar as sequências sem perder a informação da espécie correspondente.

Estratégias de Implementação

Para a realização do blast é necessário extrair as sequências presente no ficheiro da base de dados e a sequência query. De seguida é possível efetuar o blast e obter as top10 sequências mais semelhantes com a referência. Seguidamente, é realizado um alinhamento global das duas primeiras sequências, sendo progressivamente adicionadas as restantes sequências, sendo que este processo é assegurado pelo alinhamento múltiplo progressivo, uma classe implementada. Posteriormente é construída a árvore filogenética e o grafo.

Resultados

Para a obtenção das top10 sequências mais semelhantes à referência, foi implementada uma class MyBlast, com diversas funções, que se encontram descritas na Tabela 1 e uma função top10(base_dados,referencia) que permite obter uma lista com 11 listas a referência mais as 10 melhores sequências.

Tabela1- Funções implementadas na classe MyBlast.

Nome da Função	Funcionalidade
rmSequenceDB(seq)	Remove uma dada sequência (seq) do ficheiro base de dados.
build_map(query,w)	A sequência query é partida em palavras de tamanho w (por defeito o tamanho é definido como 3). De seguida é criado um dicionário onde é guardada a w-palavra e a primeira posição de inicio. Caso ocorra diversas vezes é guardada uma lista com a primeira posição de cada ocorrência.
getHits(seq,query)	Retorna uma lista de tuplos com os indices de ocorrência na sequência query e na sequência a analisar.
extendsHit(seq,hit,query)	Estende a pesquisa de hits em ambas as direções enquanto a a pontuação for superior a metade das posições extendidas. Retorna um tuplo com o indice da posição de inicio da query, o inidice da posição de inicio na sequência em análise, o tamanho do alinhamento, pontuação.
hitBestScore(seq,query)	Retorna o melhor tuplo obtido no **extendsHit.**
bestAlignment(query)	Compara a query com todas as sequências dada numa base de dados e retorna a sequência mais semelhante com a query.

Seguidamente com essa lista de listas foi realizado o alinhamento múltiplo com o auxilio das classes MultipleAlignment que vai realizando o alinhamento progressivo das sequências todas, alinhando as primeiras duas, obtendo o consensus que depois é alinhado com outra sequência e por aí em diante. A classe MyAlign que vai permitir criar uma sequência consensus, PairwiseAlignment que efetua o alinhamento global para cada par de sequências e SubstMatrix para utilizar a matriz de substituição, descritos na Tabela2.

Tabela2- Funções implementadas em cada classe

Nome da função	Funcionalidade
Class MultipleAlignment	
num_seqs()	Retorna o número de sequências
add_seq_alignment(alignment,seq)	Alignment é uma class MyALign que possui as sequências já alinhadas. É gerado um consensus, seguidamente esse consensus é alinhado com a seq introduzida. Retorna uma classe MyAlign com o novo alinhamento
align_consensus()	Realiza o primeiro alinhamento com as duas sequências da lista e de seguida vai adicionando as restantes sequências ao alinhamento uma de cada vez, chamando a função add_seq_alignment(alignment,seq)
Class MyAlign	
num_seqs()	Retorna o número de sequências já alinhadas
column(indice)	Retorna o caracter num determinado indice em todas as sequências alinhadas
consensus()	Dadas as sequências alinhadas, gera uma sequência consensus
Class PairwiseAlignment	
score_pos(c1,c2)	Pontua uma posição do alinhamento dados os caracteres das duas sequências a alinhar
score_alin(alin)	Pontua todo o alinhamento de duas sequências
needleman_Wunch(seq1,seq2)	Alinhamento global de duas sequências
recover_align()	Retorna uma Classe MyAlign com duas sequências alinhadas
Class SubstMatrix	
score_pair(c1,c2)	Retorna o valor da matriz correspondente à substituição de c1 por c2 e vice versa
read_submat_file(filename,sep)	Lê o ficheiro da matriz de substituição dando o nome do ficheiro e o separador
create_submat(match, mismatch, alphabet)	Cria uma matriz de substituição dando o alfabeto, o valor para correspondência e o valor caso contrário

Para a a formação da árvore filogenética foram implementadas as classes NumMatrix, HierarchicalClustering, UPGMA e BinaryTree. Além disso, também foi incluída a classe MyGraph para a formação do grafo, contudo, não são referidas funcionalidades visto o trabalho não ter tido sucesso até essa etapa.

Foi implementado o ficheiro runME.py de forma a executar todas as operações, sendo que para as tornar mais visuais foi criado um menu interativo.

Conclusão

Os ficheiros da base de dados e da referência foram abertos com sucesso, tendo sido realizado o blast e obtido as 10 sequências mais semelhantes à referência. Após o primeiro alinhamento (**Figura1** em anexo Primeira Lista referente ao Homo_sapiens e Pan_troglodytes), a função **align_consensus** gerou a sequência consensus. Contudo, após o alinhamento entre o consensus e a terceira sequência não foi possível criar outro consensus nem acidionar as restantes sequências ao alinhamento com a função **add_seq_alignment**, visto existirem incompatibilidades entre a as duas funções referidas acima. Visto não conseguir progredir, o trabalho apenas conta com um menu interativo, a visualização do top10 das sequências mais semelhantes à referência.

Referências

1.Needleman, Saul & Wunsch, Christian. (1970). A General Method Applicable to the search for similarities in the amino acid sequence of two proteins. Jornal Molecular Biology. 48. 443-453.

Anexos

Figura1- Print do primeiro e segundo alinhamento

```
To progessive multiple alignment we use Blosum62.mat as substitution matrix
[['Homo_sapiens', 'M--------ELSVLLFLALLTGLLLLLVQRHPNTHDRLPPGPRPLPLLGNLLQMDRRGLLKSFLRFREKYGDVFTVHLGPRPVVMLCGVEAIREALVDKAEAFSGRGKIAM
VDPFFRGYGVIFANGNRWKVLRRFSVTTMRDFGMGKRSVEERIQEEAQCLIEELRKSKGALMDPTFLFQSITANIICSIVFGKRFHYQDQEFLKMLNLFYQTFSLISSVFGQLFELFSGFLKYFPGAHRQ
VYKNLQEINAYIGHSVEKHRETLDPSAPKDLIDTYLLHMEKEKSNAHSEFSHQNLNLNTLSLFFAGTETTSTTLRYGFLLMLKYPHVAERVYREIEQVIGPHRPPELHDRAKMPYTEAVIYEIQRFSDLL
PMGVPHIVTQHTSFRGYIIPKDTEVFLILSTALHDPHYFEKPDAFNPDHFLDANGALKKTEAFIPFSLGKRICLGEGIARAELFLFFTTILQNFSMASPVAPEDIDLTPQECGVGKIPPTYQIRFLPR']
, ['Pan_troglodytes', 'MQGSQTRTMELSVLLFLALLTGLLLLLVQRHPNTHGRLPPGPRPLPLLGNLLQMDRRGLLKSFLRFREKYGDVFTVHLGPRPVVMLCGVEAIREALVDKAEAFSGRG
KIAMVDPFFRGYGVIFANGNRWKVLRRFSVTTMRDFGMGKRSVEERIQEEAQCLIEELRKSKGALMDPTFLFQSITANIICSIVFGKRFHYQDQEFLKMLNLFYQTFSLVSSVFGQLFELFSGFLKYFPG
AHRQVYKNLQEINAYIGHSVEKHRETLDPSAPKDLIDTYLLHMEKEKSDAHSEFSHQNLTLNTLSLFFAGTETTSTTLRYGFLLMLKYPHVAERVYREIEQVIGPHRPPELHDRAKMPYTEAVIYEIQRF
SDLLPMGVPHIVTQHTSFRGYIIPKDTEVFLILSTALHDPHYFEKPDAFNPDHFLDANGALKKNEAFIPFSLGKRICLGEGIARAELFLFFTTILQNFSVASPEAPEDIDLTPQECGVGKIPPTYQIRFL
PR']]
[['consensus', 'MQGSQTRTMELSVLLFLALLTGLLLLLVQRHPNTHDRLPPGPRPLPLLGNLLQMDRRGLLKSFLRFREKYGDVFTVHLGPRPVVMLCGVEAIREALVDKAEAFSGRGKIAMVDP
FFRGYGVIFANGNRWKVLRRFSVTTMRDFGMGKRSVEERIQEEAQCLIEELRKSKGALMDPTFLFQSITANIICSIVFGKRFHYQDQEFLKMLNLFYQTFSLISSVFGQLFELFSGFLKYFPGAHRQVYK
NLQEINAYIGHSVEKHRETLDPSAPKDLIDTYLLHMEKEKSNAHSEFSHQNLNLNTLSLFFAGTETTSTTLRYGFLLMLKYPHVAERVYREIEQVIGPHRPPELHDRAKMPYTEAVIYEIQRFSDLLPMG
VPHIVTQHTSFRGYIIPKDTEVFLILSTALHDPHYFEKPDAFNPDHFLDANGALKKTEAFIPFSLGKRICLGEGIARAELFLFFTTILQNFSMASPVAPEDIDLTPQECGVGKIPPT--------'], [
'Pan_paniscus', 'M--------ELSVLLFLALLTGLLLLLVQRHPNTHGRLPPGPRPLPLLGNLLQMDRRGLLKSFLRFREKYGDVFTVHLGPRPVVMLCGVEAIREALVDKAEAFSGRGKIAMVD
PFFRGYGVIFANGNRWKVLRRFSVTTMRDFGMGKRSVEERIQEEAQCLIEELRKSKGALMDPTFLFQSITANIICSIVFGKRFHYQDQEFLKMLNLFYQTFSLVSSVFGQLFELFSGFLKYFPGAHRQVY
KNLQEINAYIGHSVEKHRETLDPSAPKDLIDTYLLHMEKEKSDAHSEFSHQNLTLNTLSLFFAGTETTSTTLRYGFLLMLKYPHVAERVYREIEQVIGPHRPPELHDRAKMPYTEAVIYEIQRFSDLLPM
GVPHIVTQHTSFRGYIIPKDTEVFLILSTALHDPHYFEKPDAFNPDHFLDANGALKKNEAFIPFSLGKRICLGEGIARAELFLFFTTILQNFSVASPEAPEDIDLTPQECGVGKIPPTYQIRFLPR']]
```