Esercitazione 24-05-2011 laboratorio Algoritmi e Linguaggi per Bioinformatica: Matlab – Bioinformatics Toolbox

Reference: Bioinformatics toolbox 3 user guide:
http://www.mathworks.com/help/pdf_doc/bioinfo/bioinfo_ug.pdf
Solutions to the following exercises can be found at page 2.2

Exercise 1: Sequence statistics (solution at page 2.2 of the user guide)

After sequencing a piece of DNA, one of the first tasks is to investigate the nucleotide content in the sequence. Starting with a DNA sequence, this example uses sequence statistics functions to determine mono-, di-, and trinucleotide content, and to locate open reading frames.

a. Reading Sequence Information

The consensus sequence for the **human mitochondrial genome** has the GenBank accession number NC_012920. Since the whole GenBank entry is quite large and you might only be interested in the sequence, you can get just the sequence information.

1. *Get sequence information from a Web database.* For example, retrieve sequence information for the human mitochondrial genome, in the MATLAB Command Window [function getgenbank].

2. If you don't have a Web connection, you can load the data from a MAT file included with the Bioinformatics Toolbox software, using the command *load mitochondria* The load function loads the sequence mitochondria into the MATLAB.

3. Get information about the sequence [function whos].

b. Determining Nucleotide Composition

The following procedure illustrates how to determine the monomers and dimers, and then visualize data in graphs and bar plots. Sections of a DNA sequence with a high percent of A+T nucleotides usually indicate intergenic parts of the sequence, while low A+T and higher G+C nucleotide percentages indicate possible genes. Many times high CG dinucleotide content is located before a gene.

1. Plot monomer densities and combined monomer densities in a graph [function ntdensity].

2. Count the nucleotides [function basecount].

3. Count the nucleotides in the reverse complement of a sequence [function seqrcomplement].

4. Visualize the nucleotide distribution [function basecount]

5. Count the dimers in a sequence and display the information in a bar chart [function dimercount]

c. Determining Codon Composition

The following procedure illustrates how to look at codons for the six reading frames. Trinucleotides (codon) code for an amino acid, and there are 64 possible codons in a nucleotide sequence. Knowing the percent of codons in your sequence can be helpful when you are comparing with tables for expected codon usage.

1. Count codons in a nucleotide sequence [function codoncount]

2. Count the codons in all six reading frames and plot the results in heat maps [functions figure, subplot, codoncount. Use a for cycle for iterating reading frames. See reference user guide.]

d. Open Reading Frames

The following procedure illustrates how to locate the open reading frames using a specific genetic code. Determining the protein-coding sequence for a eukaryotic gene can be a difficult task because introns (noncoding sections) are mixed with exons. However, prokaryotic genes generally do not have introns and mRNA sequences have the introns removed. Identifying the start and stop codons for translation determines the protein-coding section, or open reading frame (ORF), in a sequence. Once you know the ORF for a gene or mRNA, you can translate a nucleotide sequence to its corresponding amino acid sequence.

1. Display open reading frames (ORFs) in a nucleotide sequence [function seqshoworfs].

2. Display ORFs using the Vertebrate Mitochondrial code [function seqshoworfs. Notice: this function returns a *structure* data type].

3. Find the corresponding stop codon [function find. Use structure dot notation to access start and stop indexes. See user guide for more information].

4. Using the sequence indices for the start and stop of the gene, extract the subsequence from the sequence.

5. Determine the codon distribution [function codoncount].

6. Look up the amino acids for codons ATA, CTA, ACC, and ATC [function aminolookup].

e. Amino Acid Conversion and Composition

The following procedure illustrates how to extract the protein-coding sequence from a gene sequence and convert it to the amino acid sequence for the protein. Determining the relative amino acid composition of a protein will give you a characteristic profile for the protein. Often, this profile is enough information to identify a protein. Using the amino acid composition, atomic composition, and molecular weight, you can also search public databases for similar proteins.

1. Convert a nucleotide sequence to an amino acid sequence. Only the protein-coding sequence between the start and stop codons should is converted. The Vertebrate Mitochondrial genetic code should be employed for conversion [function nt2aa].

2. Compare your conversion with the published conversion in the GenPept database [function getgenpept].

3. Count the amino acids in the protein sequence [function aacount].

4. Determine the atomic composition and molecular weight of the protein [functions atomiccomp, molweight].

Exercise 2: Sequence Alignment (solution at page 2.22 of the user guide)

Determining the similarity between two sequences is a common task in computational biology. Starting with a nucleotide sequence for a human gene, this example uses alignment algorithms to locate and verify a corresponding gene in a model organism.

a. Retrieving Sequence Information from a Public Database

The following procedure illustrates how to find the nucleotide sequence for a human gene in a public database and read the sequence information into the MATLAB environment. Many public databases for nucleotide sequences (for example, GenBank, EMBL-EBI) are accessible from the Web. The MATLAB Command Window with the MATLAB Help browser provide an integrated environment for searching the Web and bringing sequence information into the MATLAB environment.

1. Get sequence data into the MATLAB environment. For example, to get sequence information for the **human gene HEXA** (accession number 'NM_000520') [function getgenbank]

b. Searching a Public Database for Related Genes

The following procedure illustrates how to find the nucleotide sequence for a ***mouse gene*** related to a ***human gene***, and read the sequence information into the MATLAB environment. The sequence and function of many genes is conserved during the evolution of species through homologous genes. Homologous genes are genes that have a common ancestor and similar sequences. One goal of *searching a public database is to find similar genes*. If you are able to locate a sequence in a database that is similar to your unknown gene or protein, it is likely that the function and characteristics of the known and unknown genes are the same.

1. Get sequence information for the mouse gene into the MATLAB environment (accession number 'AK080777').

c. Locating Protein Coding Sequences

The following procedure illustrates how to convert a sequence from nucleotides to amino acids and identify the open reading frames. A nucleotide sequence includes regulatory sequences before and after the protein coding section. By analyzing this sequence, you can determine the nucleotides that code for the amino acids in the final protein.

After you have a list of genes you are interested in studying, you can determine the protein coding sequences. This procedure uses the human gene HEXA and mouse gene HEXA as an example.

1. If you did not retrieve gene data from the Web, you can load example data from a MAT-file included with the Bioinformatics Toolbox software (command load hexosaminidase). The structures humanHEXA and mouseHEXA load into the MATLAB.

2. Locate open reading frames (ORFs) in the human gene [function seqshoworfs]. Seqshoworfs creates a structure containing the position of the start and stop codons for all open reading frames (ORFs) on each reading frame.

3. Locate open reading frames (ORFs) in the mouse gene [function seqshoworfs]. The mouse gene shows the longest ORF on the first reading frame.

d. Comparing Amino Acid Sequences

The following procedure illustrates how to use global and local alignment functions to compare two amino acid sequences. You could use alignment functions to look for similarities between two nucleotide sequences, but alignment functions return more biologically meaningful results when you are using amino acid sequences.

After you have located the open reading frames on your nucleotide sequences, you can convert the protein coding sections of the nucleotide sequences to their corresponding amino acid sequences, and then you can compare them for similarities.

1. Using the open reading frames identified previously, convert the human and mouse DNA sequences (first reading frame) to the amino acid sequences [function nt2aa].

2. Draw a dot plot comparing the human and mouse amino acid sequences [function seqdotplot]. The diagonal line shown below indicates that there may be a good alignment between the two sequences.

3. Globally align the two amino acid sequences, using the Needleman-Wunsch algorithm [functions nwalign, showalignment]. Notice that the calculated identity between the two sequences is 60%. The alignment is very good between amino acid position 69 and 599, after which the two sequences appear to be unrelated. Notice that there is a stop (*) in the sequence at this point. If you shorten the sequences to include only the amino acids that are in the protein you might get a better alignment. Include the amino acid positions from the first methionine (M) to the first stop (*) that occurs after the first methionine.

4. Find the indices for the stops in the sequences [function find]. Looking at the amino acid sequence for humanProtein, the first M is at position 70, and the first stop after that position is actually the second stop in the sequence (position 599). Looking at the amino acid sequence for mouseProtein, the first M is at position 11, and the first stop after that position is the first stop in the sequence (position 557).

5. Truncate the sequences to include only amino acids in the protein and the stop.

6. Globally align the trimmed amino acid sequences [function nwalign, showalignment]. Notice that the percent identity for the untrimmed sequences is 60% and 84% for trimmed sequences.

7. Another way to truncate an amino acid sequence to only those amino acids in the protein is to first truncate the nucleotide sequence with indices from the *seqshoworfs* function (see the user's guide).

8. Locally align the two amino acid sequences using a Smith-Waterman algorithm [function swalign].

9. Show the alignment in color.