MIDDLE LAYER API

Gabriele Nocchi

g.nocchi@hotmail.com

queries.pm

Function	Arguments	Returns
get_results	2 strings: - The type of search (ID, location, product or accession) from the dropdown box of the application. - The search parameter typed by the user in the search box of the application.	Hash of results: - Keys = Genebank Accessions. - Values = ID + Location + Prod. Name.
get_sequences	1 string: the Genebank Accession number of the entry selected by the user.	3 strings: - The DNA sequence. - The translated amino acid sequence. - The codon start position (ie. The reading frame).
make_exons_hash	1 string: the Genebank Accession number of the entry selected by the user.	1 hash: - Keys = Exons start positions. - Values = Exons length.

cal.pm

Function	Arguments	Returns
protein_spacing	1 string: the amino acids sequence.	1 sequence: the amino acids sequence with a “-“ sign on either side of each amino acid (for presentation purposes in the front end).
connect_exons	2 strings and 1 hash: - The full DNA sequence. - The codon start position. - The exons positions hash.	1 string: - The coding sequence (connected exons).
extract_exons	1 string and 1 hash: - The full DNA sequence. - The exons positions hash.	1 array where each value in the array is an exon sequence.

codons.pm

Function	Arguments	Returns
calc_cod_freq	1 string: the coding sequence.	1 hash: - Keys = Codons. - Values = Frequencies.
calc_cod_ratio	2 strings: - The coding sequence. - The amino acid sequence.	1 hash: - Keys = Codons. - Values = Ratios.
map_codons	2 strings: - The coding sequence. - The amino acid sequence.	1 hash: - Keys = Codons. - Values = Corresponding Amino Acids.

enzymes.pm

Function	Arguments	Returns
get_regions	1 string and 1 hash: - The full DNA sequence. - The exons positions hash.	3 strings: - The 5 region. - The middle region. - The 3 region.
check_ecori check_bamhi check_bsumi	3 strings: - The 5 region. - The middle region. - The 3 region.	True if the enzyme can cut the sequence in the 5 and/or 3 region but not in between. False if the opposite.
get_complementary	1 string: the restriction enzyme recognition motif typed by the user in the interface.	1 string: the complementary sequence of the motif the user typed in the interface.
check_enzyme	5 strings: - The 5 region. - The middle region. - The 3 region. - The restriction enzyme recognition motif entered by the user. - The complementary sequence of the restriction motif entered by the user.	True if the enzyme recognition motif entered by the user is found in the 5 and/or 3 region of the DNA sequence but not in between. False if the opposite.

The middle layer of the application is composed of 4 modules:

- **queries.pm** is the Data Access Tier. This module contains the SQL necessary to retrieve the data from the database.
- **cal.pm** is part of the Business Logic Tier. This module contains subroutines which perform pre-calculations on the data

retrieved from the database (such as building the coding sequence using exons data) before the data is passed to the front-end for presentation or to functions of the other 2 modules for restriction enzymes and codons calculations.

- **codons.pm** is part of the Business Logic Tier. This module contains subroutines which perform codon frequencies calculations.
- **enzymes.pm** is part of the Business Logic Tier. This module contains subroutines for assessing whether the 3 given restriction enzymes or an enzyme entered by the user can cut the DNA sequence of a gene at the 5 and/or 3 region but not in between.

To install the middle layer modules should be saved within a directory and should be used within the cgi scripts of the application.

To use the modules, "use lib" followed by the pathway name (depending on the location of the modules) should be used in the cgi scripts.

2 scripts, **extract.pl** and **chromosome.pl**, are also included in the middle layer.

Those 2 scripts are required for the pre-calculation of the static table of codons frequencies of the whole chromosome.

extract.pl is used to extract and concatenate all coding sequences in the database. This script creates a file with the concatenated sequences on the first line.

chromosome.pl is used on the file created with **extract.pl**. This script carries the codons frequencies calculations on the concatenated coding sequences. This script creates a file which has the codons data for the whole chromosome already indented within html table tags agreed with the front end.