Frontend documentation

The frontend of the application comprises 3 HTML scripts and 3 CGI scripts.

HTML scripts:

1. Projhome.html – The homepage, which provides the user with the ability to search for a gene in the database. As well as a link to obtain instructions on website use and also an 'about' page.
2. Aboutus.html – Detailing the outlining structure of the website application and those responsible for it.
3. Instructions.html – Briefly explaining how to search through the database and obtain the results desired.

CGI scripts:

1. Projcgi1.pl – Displays the initial search results requested by the user
2. Projcgi2.pl – Displays details about the gene selected by the user in the initial results page, namely coding regions highlighted, coding sequence aligned with amino acid sequence, amino acid usage frequency, amino acid usage ratio and the option of finding out if any given restriction enzyme sequence can cut at specific points in the chosen gene sequence.
3. Projcgi3.pl – Displays the result of whether the user's restriction enzyme is able to cut at specific points in the gene sequence.

Tabular synopsis of CGI scripts

Projcgi.pl		
Desired user interface	Final argument(s)	Unpacking method
Initial results page of genes with 4 identifiers in tabular form.	1 hash called %results collected via a function in the queries.pm module.	Tabular form using a regex to pluck out specific IDs from the value of the %results hash and present them in separate cells of the initial results table

Projcgi2.pl

Desired user interface	Final argument(s)	Unpacking method
To display DNA sequence of gene with highlighted coding regions.	1 array called @exons collected via a function in the cal.pm module and is renamed to @exon_box in this CGI script.	Bolden the coding regions in the entire sequence and store in a string. Then use the perl unpack function to print the desired amount of nucleotides per line i.e. 50.
To display coding regions alongside amino acid sequence.	2 strings required called the $spaced_seq and $coding_seq which is renamed to $upper_seq to raise the string to uppercase in this CGI script. Both strings are collected via functions in the cal.pm module.	To display a specific amount of nucleotides per line and to have the coding sequence alongside the amino acid sequence. A while loop was formed to move along both strings and extract the desired amount of nucleotides before a newline is printed, this is repeated until both strings are completely printed.
To display coding usage frequencies and coding usage ratios in tabular form.	3 hashes called %codon_freq, %codon_ratio, %translation which are collected via functions in the codons.pm module.	Using a table composed of 4 columns: amino acid, codon, codon frequency and codon ratio.
To affirm or disaffirm whether the sequence of the gene in question can be cut at the 5' or 3' end of the sequence, but not in between by using either ECORI, BAMHI or BSUMI.	3 strings called $five_end, $middle_sect, $three_end which are collected via functions in the enzymes.pm module.	3 simple 'If' statements printing a statement affirming or disaffirming whether each of the specified restriction enzymes will effectively cut the sequence in question.

Projcgi3.pl		
Desired user interface	Final argument(s)	Unpacking method
To display a statement delineating whether the user's restriction enzyme sequence works on the gene or not.	5 strings called $five_end, $middle_sect, $three_end, $motif, $complem_motif are called on using functions in the queries.pm and enzymes.pm module.	A simple 'if' statement printing a statement affirming or disaffirming the restriction enzyme usability.