Code Documentation BL Layer
Author: Shahid Hadi
BL layer has two sets of functions:
First: Search functions that work as a bridge between the front end and the database layer. These functions only return the DB layer functions without any intervention. The following functions are in thish category: getAllEntries(), searchGeneID(geneID), searchProteinProduct(proteinProductString), searchAccession(accessionCode), searchChromLocation(chromosomeLocation).

Second: Functions that perform various calculations on the database and return the results to the front end. These functions can be further divided into two groups:

a- Functions that perform calculation over the whole database once and return the results. There is only on function in this category, getGeneralCodonFreq(), which takes no input and it applies the code written for a single gene codon frequency to the whole genes in chromosome nine ONLY ONCE and saves the results of the run as a dictionary in the return part of the function. Returns a dictionary with the keys being codes of the amino acids, like 'Ala(A)', each value consists of three lists:
    1 - List of codons used by the key amino acid.
    2 - List containing corresponding percentage usage for each codon.
    3 - List containing corresponding frequencies for each codon. """
  return ({'Ala(A)': [('gct', 'gcc', 'gca', 'gcg'), [0.5, 1.35, 2.82, 1.75], [0.08, 0.21, 0.44, 0.27]]}

b- Functions that perform calculations over a single gene. These functions take the accession code of the gene, as a string or a variable assigned to the string, pickup all the information need for calculations in BL layer from the DB layer using a single DB layer function, dbapi.getAllGeneInfo(accession), perform their calculations and return results to the front end.

   a. getDNAseqAndCodingRegions(accession): This function generates the coding sequence and shows its exons as lists within a list that contains the entire gene sequence. The coding exons are represented as lists containing a single string, and non-coding sequences are string objects inside the returned list by the function.
   Returns([‘cggttaagc’, [‘atgacggggctggc’], ‘ccggttacgta’])

   b. alignNucleotide_to_aminoacid(accession): This function aligns each amin acid in the protein sequence (single letter amino acid codes are used) to their corresponding codons in the coding sequences. Returns a list, with each instance in the list being a string containing the amino acid code separated from its corresponding sequence by a colon.
   Returns(['M:atg', 'G:ggt', 'F:ttt', 'L:tta'])

   c. RE_sites_list_cat(accession): This function returns the restriction enzyme sites for 5 enzymes; EcoRI, BamHI, BsuMI, HindIII, and StuI, and categorize them according to their cutting sites and applicability for gene cloning purposes. Returns two dictionaries:
      1- First dictionary classifies the enzymes according to their availability and applicability. it has three lists:

1- 'available_at_5p': should contain all the enzymes that cut at 5' end and DOES NOT cut at the coding region.
2- 'available_at_3p': should contain all the enzymes that cut at 3' end and DOES NOT cut at the coding region.
3- not_applicable: should contain all the enzymes that EITHER cut within the coding reion OR does not have any RE sites within the DNA sequence of the gene of interest

2- Second dictionary links each enzyme to all the sites that it matches within the gene of interest by the first index of the match.

Returns ({'available_at_5p': [], 'available_at_3p': [], 'not_applicable': ['EcoRI', 'BamHI', 'BsuMI', 'HindIII', 'StuI']}
{'EcoRI': [543, 1484, 1432, 6191, 9597, 9870],
'BamHI': [8128, 1073, 3004],
'BsuMI': [6043],
'HindIII': [533, 734, 8953, 11430, 12179],
'StuI': []})

d. input_seq_RE_sites(accession, your_choice): This function works exactly as the previous function. And the only difference is that it takes two inputs. The second input is a sequence for a custom restriction enzyme entered by the user. So, this function returns results for the custom function rather than the predefined Res.

e. geneCodonUsage(accession): This function This function calculates the codon percentage usage, ratio of a codon usage per 100 codons, and codon frequencies, the ratio of each codon among codons used by a certain amino acid, and returns a list to show the difference between percentage codon usage of the entire chromosome 9 and the gene of interest, the list returns 1 if the difference is more than two folds and 0 if less than two folds. Returns a dictionary with the keys being codes of the amino acids, like 'Ala(A)', each value consists of three lists:

1 - List of codons used by the key amino acid.
2 - List containing corresponding percentage usage for each codon.
3 - List containing corresponding frequencies for each codon.

And a list that holds the values of the statistical test for each corresponding codon.

Returns ({'Ala(A)': [('gct', 'gcc', 'gca', 'gcg'), [0.97, 0.24, 0.97, 0.0], [0.44, 0.11, 0.44, 0.0]]}