Introduction to Bioinformatics
Problem Set 5: Genome Analysis Investigations

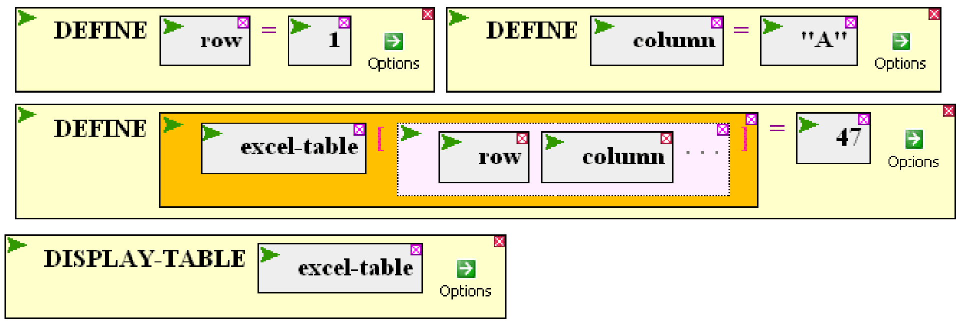
DEFINE-FUNCTION

1. Define a function that accepts a number and returns its square.
2. Define a function that accepts two words and displays them in reverse order.
3. Define a function that accepts a DNA sequence and returns a palindrome that begins with the given sequence. You'll want to know about the INVERSION-OF function in the STRING/SEQUENCE, String-production menu.

Tables and the [ ] notation

4. Create a table of the form letter[number], where letter[1] = "A", letter[2] = "B", etc.

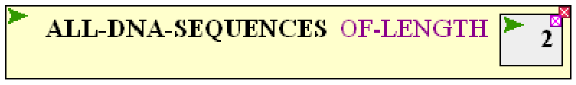
5. Make and display an x * y multiplication table for x and y, each going from 1 to 15. The [] function (in the List/Tables menu) will be essential here. You can use it to define elements of a two-dimensional table, as shown by example below:



```
DEFINE row = 1
DEFINE column = "A"
DEFINE excel-table [ row column . . . ] = 47
DISPLAY-TABLE excel-table
```

6. Make and display a table containing information about the organisms known to BioBIKE. The row labels should be the names of the organisms. The column labels should genome size, number of genes, and GC-fraction.

7. Determine the frequency of each dinucleotide in the genome of ss120. Put the values in a table. The following function will be useful:



```
ALL-DNA-SEQUENCES OF-LENGTH 2
```

Statistics

8. In which of the following cases would a **chi-square test** be useful? In which would a **t-test** be useful?
   a. Are the genes of *Anabaena* PCC 7120 longer, on average, than the genes of ss120?
   b. Mendel counted 705 purple flowers and 224 white flowers. Is this reasonably close to a 3:1 ratio?
   c. Your unidentified viral sequence has dinucleotide counts of {AA = 60, AC = 72, AG = 52, …}. Could the fragment reasonably have been derived from the virus Mx8?
   d. Is expression of the gene encoding melanin induced by ultraviolet radiation? I've measured expression of the gene 12 times: 6 times with UV and 6 times without.

9. What is the average size of the genes of *Prochlorococcus marinus* ss120? Of *Anabaena* PCC 7120? Are the genes of ss120 significantly smaller than the genes of A7120? Answer the question by doing a t-test. Note that there exists a T-TEST function, living in the ARITHMETIC / Statistics menu. Answer the question also by doing a simulation.

10. Is Mendel's results reasonably described as 3:1? Answer the question by means of a chi-square test (note there does ***not*** exist a chi-square function in BioBIKE!) and also by doing a simulation.

11. Return to *What is a gene*, part C, and construct by hand the table constructed in the investigation by MAKE-PSSM-FROM.

12. What is the frequency of 6-nt palindromic sequences in *Anabaena* PCC 7120?

    a. Construct a form that provides all palindromic sequences of length 6.

    b. Count all palindromic sequences of length 6 in *Anabaena* PCC 7120.

    c. Compare the incidence of each with the predicted number of counts.

    d. Calculate for each the chi-squared score, comparing predicted with observed counts, then create and sort a list of palindromic sequences by its chi-squared score.

13. Construct a codon frequency table for the virus myxococcus_phage_mx8 as described by Karlin (2001) [Trends in Microbiology 9:335-343]. In brief the table should have the following format (shown by example):

    Codon-frequency["GGC"]

    where the frequency for each codon is given as the ***fraction*** of times the codon is used from amongst all the codons encoding the same amino acid. Thus, the above example would be:

$$\frac{\text{(counts-of "GGC")}}{\text{(counts-of "GGA" + counts-of "GGC" + counts-of "GGG" + counts-of "GGT")}}$$

    since GGA, GGC, GGG, and GGT all encode glycine.

    Let's break up the task:

    <u>Create `codon-count-table`</u>

    a. Generate a list of all codons in the genes of myxococcus_phage_mx8. You'll want to consider all the GENES-OF the phage, all the SEQUENCES-OF the genes, and then SPLIT the sequences every three nucleotides. Finally, you'll want to SIMPLIFY the resulting lists of lists ((…) (…) (…)…) into a single list.

    b. Define a **list** (call it `codon-count-list`) that contains the counts of each of the 64 triplet sequences amongst the codons of myxococcus_phage_mx8. Of course you'll want to make use of the COUNTS-OF and ALL-DNA-SEQUENCES functions, plus the result you obtained in **part a**.

    c. Learn how to create a single **element of a table** (call the table `codon-count-table`). Use DEFINE and [ ] functions to do this. For example:

    (DEFINE codon-count-table[*any codon in quotes*] AS *any number of your choice*)

d. Fill in `codon-count-table` with counts for all 64 triplet sequences. Use APPLY-FUNCTION-OF, using the function:

```
Function-of (codon codon-count)
     = (DEFINE codon-count-table[codon] = codon-count)
```

and then apply that function of codon and codon-count to a list of codons and a list of codon counts. Note that you made a list of codon counts in part ***a***, where you also used a list of codons. If your table has been defined properly, then you should be able to do things like this:

```
codon-count-table["AAA"]
```

and get the counts for that codon as the result (which should be 43).

Write a loop to calculate entries for `codon-frequency-table`

e. Write a ***loop*** that considers in turn each of the 20 amino acids and displays the name of the amino acid

f. Modify the loop so that you create a loop variable (call it **`codons`**) that you set equal (each iteration) to a list of codons that encode the specific amino acid. You may have to click the **reveal-all** option in the main menu of FOR-EACH. You'll be interested in a function AA-TO-CODONS that exists in the Genes-Proteins menu, Translation submenu. Display that list alongside the name of the amino acid.

g. Modify the loop so that you create a loop variable (call it **codon-counts**) that you set equal (each iteration) to a list of codon counts for the codons you defined in **part f**. You'll want to make use of the function ELEMENTS-OF-TABLE. In place of *index* put the list of codons for the amino acid. Display the contents of this variable alongside the name of the amino acid and the list of codons.

h. Modify the loop so that you create a loop variable (call it **count-sum**) that you set equal (each iteration) to the sum of the codon counts that you defined in **part g**. SUM-OF will be a useful function. Display the contents of this variable alongside all the other things you're already displaying.

i. Modify the loop so that you create a loop variable (call it **codon-frequencies**) that you set equal (each iteration) to the codon-counts divided by the count sum. This should give you a list of frequencies that add up to one. Display this new variable alongside the others and confirm that the frequencies do add up to 1.

j. Now that you have the frequencies, all that's left is to enter each of the frequencies into a table called **codon-freq-table** within the body of the loop. You'll do this in very much the same way as you created **codon-count-table** in **part d**.

k. See what you got, using the DISPLAY-TABLE